

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-200221

(P2000-200221A)

(43)公開日 平成12年7月18日(2000.7.18)

(51)Int.Cl.<sup>7</sup>

G 0 6 F 12/12

識別記号

F I

C 0 6 F 12/12

テーマコード(参考)

A

D

審査請求 有 請求項の数 9 O L (全 23 頁)

(21)出願番号 特願平11-285512

(22)出願日 平成11年10月6日(1999.10.6)

(31)優先権主張番号 特願平10-311044

(32)優先日 平成10年10月30日(1998.10.30)

(33)優先権主張国 日本(J P)

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 斎藤 靖彦

東京都港区芝五丁目7番1号 日本電気株式会社内

(74)代理人 100096231

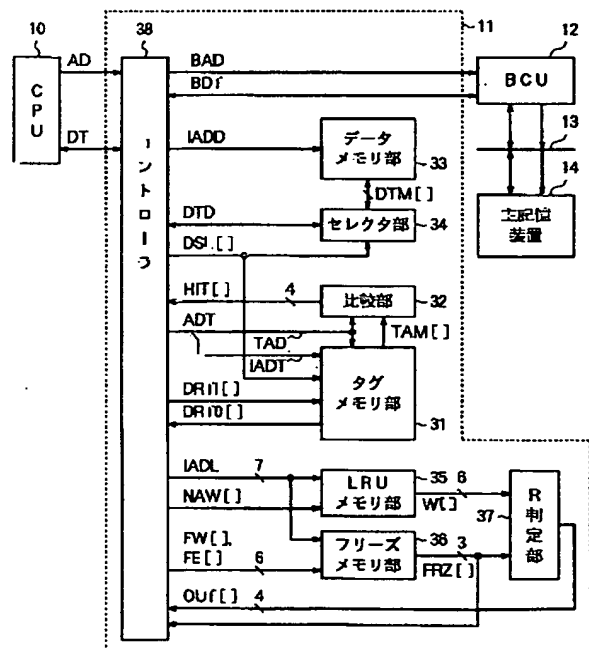
弁理士 稲垣 清

(54)【発明の名称】 キャッシュメモリ装置及びその制御方法

(57)【要約】

【課題】 簡素な回路構成を備えながらも、キャッシュメモリのスループットを低下させることなく2以上のウェイを対象に円滑なフリーズ処理が実行でき、ニーズに対応したキャッシュメモリ機能を実現できるキャッシュメモリ装置及びその制御方法を提供する。

【解決手段】 LRUデコード方式を採用し、Nを2以上の整数としたNウェイ・セット・アソシアティブ方式のキャッシュメモリ装置11は、キャッシュメモリの各ウェイのアクセス履歴情報を記憶するLRUメモリ部35と、各ウェイのフリーズ情報を記憶するフリーズメモリ部36と、記憶されたアクセス履歴情報を修正することなく該アクセス履歴情報とフリーズ情報とに基づいて、特定のウェイをフリーズするためのリブレース信号を生成するリブレース判定部37とを備えている。



## 【特許請求の範囲】

【請求項1】 LRUデコード方式を採用し、Nを2以上の整数としたNウェイ・セット・アソシアティブ方式のキャッシュメモリ装置において、キャッシュメモリの各ウェイのアクセス履歴情報を記憶するLRUメモリと、各ウェイのフリーズ情報を記憶するフリーズメモリと、記憶されたアクセス履歴情報を修正することなく該アクセス履歴情報と前記フリーズ情報とに基づいて、特定のウェイをフリーズするためのリプレース信号を生成するリプレース判定手段とを備えることを特徴とするキャッシュメモリ装置。

【請求項2】 前記リプレース判定手段が、前記フリーズ情報と前記アクセス履歴情報とを取り込んで中間情報を生成するフリーズ制御回路と、前記フリーズ制御回路からの前記中間情報に対応して前記リプレース信号を生成するLRUデコード回路とを備えることを特徴とする請求項1に記載のキャッシュメモリ装置。

【請求項3】 前記フリーズ情報としてN-1ウェイに対応する記憶容量を備えることを特徴とする請求項1又は2に記載のキャッシュメモリ装置。

【請求項4】 前記リプレース信号が、各ウェイに夫々対応して出力されるビット値から成ることを特徴とする請求項1乃至3の内の何れか1項に記載のキャッシュメモリ装置。

【請求項5】 LRUデコード方式を採用し、Nを2以上の整数としたNウェイ・セット・アソシアティブ方式のキャッシュメモリ装置を制御する制御方法において、キャッシュメモリの各ウェイのアクセス履歴情報を記憶し、キャッシュメモリの各ウェイのフリーズ情報を記憶し、記憶された前記アクセス履歴情報を修正することなく該

アクセス履歴情報と前記フリーズ情報とに基づいてリプレース対象のウェイを算出することを特徴とするキャッシュメモリ装置の制御方法。

【請求項6】 前記アクセス履歴情報と前記フリーズ情報とに基づいてリプレース対象のウェイを算出する際に、前記フリーズ情報に対応して前記アクセス履歴情報を中間情報に変換した後にリプレース対象のウェイを算出することを特徴とする請求項5に記載のキャッシュメモリ装置の制御方法。

【請求項7】 前記アクセス履歴情報と前記フリーズ情報とに基づいてリプレース対象のウェイを算出する際に、Nウェイのアクセス順序を最近アクセスされたウェイから順に並べたとき、フリーズ対象のウェイが1乃至N-1のいずれかのアクセス順序になるような中間情報を生成した後にリプレース対象のウェイを算出することを特徴とする請求項5に記載のキャッシュメモリ装置の制御方法。

【請求項8】 ウェイAに対するウェイBのアクセス順序を表すアクセス履歴情報を $W[A, B]$ とし、ウェイA及びウェイBのフリーズ情報を夫々 $FRZ[A]$ 及び $FRZ[B]$ とすると、前記中間情報としてのM

$[A, B]$  への変換が、  
 $M[A, B] = W[A, B] + FRZ[A]$  (但し、 $0 \leq A < B = (N-1)$ )

$M[A, B] = (W[A, B] + FRZ[A]) \times !FRZ[B]$

(但し、 $0 \leq A < B < (N-1)$  であり、 $!FRZ[]$  は $FRZ[]$  の反転値を示す) に基づいて行われることを特徴とする請求項6又は7に記載のキャッシュメモリ装置の制御方法。

【請求項9】 リプレース対象のウェイをXとすると、該ウェイXの算出は、次式

【数1】

$$OUT[X] = \prod_{j=0}^{X-1} M[j, X] \times \prod_{j=X+1}^{N-1} !M[X, j]$$

(但し、 $0 \leq X, j \leq (N-1)$ 、 $[, ]$  内の数字で左側の数<右側の数)

に基づいて行われることを特徴とする請求項5乃至8の内の何れか1項に記載のキャッシュメモリ装置の制御方法。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、キャッシュメモリ装置及びその制御方法に関し、特に、フリーズ機能を改善したキャッシュメモリ装置及びその制御方法に関する。

【0002】

【従来の技術】コンピュータシステムの分野における大容量の主記憶装置は、その動作速度がマイクロプロセッ

サの動作速度に比して遅い。このため、最近のマイクロプロセッサでは、例えばSRAM (Static Random Access Memory) 等から成る小容量で高速なキャッシュメモリをマイクロプロセッサの内部、もしくはその近傍に配置し、データの一部をキャッシュメモリに記憶することによって、マイクロプロセッサの速度を低下させることなく作動させている。

【0003】上記コンピュータシステムでは、中央処理装置からキャッシュメモリへのリードアクセス又はライトアクセスがミスヒットした場合に、主記憶装置から新たに読み出したデータの一部が、エントリ (登録項目) としてキャッシュメモリの空きブロックに格納される。

このとき、空きブロックが存在しない場合には、複数のブロックのいずれか1つを選択し、選択されたブロックに格納されているエントリを主記憶装置に戻してブロック内を空き状態にし、この空きブロックに新たに読み出したデータを格納するエントリ置換処理が必要になる。

【0004】上記エントリ置換処理では、最も以前に参照したデータを格納しているブロックを選択する手法、即ち、LRU (Least Recently Used) デコード方式が一般的に採用されている。このLRUデコード方式によってキャッシュメモリの使用効率が向上し、その結果、マイクロプロセッサの実行速度が向上する。

【0005】しかし、マイクロプロセッサが処理するプログラムの中には、アクセス頻度は少ないものの、ひとたび起動された場合には高速に処理しなければならないような特殊な処理も存在する。例えば、自動車の制御用にマイクロプロセッサを使用し、表示パネルの制御やブレーキの制御を行う場合を考える。

【0006】表示パネルの制御は、運転中であれば速度や走行距離などの状況が時々刻々と変化するので、絶えず所定のプログラムが実行される。これに対して、急ブレーキをかけたときにタイヤがロックされないように制御するプログラムは、それほど頻繁に処理されることはない。従来のキャッシュメモリを使用した場合、表示用プログラムはキャッシュメモリに取り込まれるが、ブレーキ制御用プログラムは取り込まれている確率が低くなる。このため、急ブレーキをかけても、ブレーキ制御用プログラムの実行開始までに時間がかかることが予想される。

【0007】一方、走行距離を表示するプログラムは、実行に多少時間がかかっても問題とならないにも拘わらず、キャッシュメモリを占有することになり、キャッシュメモリの利用効率を悪くし、或いは、プログラム全体の処理速度を低下させるという問題を引き起こす。

【0008】上記問題を解決するため、キャッシュメモリにフリーズ機能やパージ機能を設けることが知られている。フリーズ機能は、アクセス頻度は少ないものの、ひとたび起動された場合には高速に処理しなければならないようなプログラムを予めキャッシュメモリ内にコピーしておき、その領域を書き換え禁止にしておく機能である。この機能を有することで、コンピュータシステムは、必要なときに所定のプログラムをキャッシュメモリから読み出して実行することができ、これにより実行時間が短縮する。

【0009】逆に、パージ機能は、アクセス頻度は多いが、実行速度がそれほど要求されないようなプログラムをキャッシュメモリ内に保存しておくことなく、その領域を解放する機能である。この機能を有することで、キャッシュメモリに余裕ができ、優先度の高い他のプログラムやデータをキャッシュメモリに取り込むことができ、これにより、キャッシュメモリの利用効率が向上

し、総合的な実行時間が短縮する。

【0010】従来のセット・アソシアティブ・キャッシュで、1ウェイのみのフリーズを可能にした半導体記憶装置が、特開平6-110787号公報に記載されている。図13は、この公報に対応する半導体記憶装置のフリーズ制御部分の構成を示すブロック図である。

【0011】図13に示すように、フリーズ制御部分は、タグメモリ51、データメモリ52、パージ・フリーズレジスタ43、LRUパージ・フリーズ制御部42、LRUメモリ優先順位情報更新制御部44、LRUメモリ40、アドレスコンパレータ57、制御部58、及びセクタ59を備えている。

【0012】タグメモリ51は、ウェイ51A~51Dを有し、キャッシュデータのキャッシュメモリにおける記憶アドレスをそのデータとして記憶するもので、本来のタグデータの他にバリッドビットが追加されている。データメモリ52は、ウェイ52A~52Dを有し、キャッシュデータを記憶するものであり、高速アクセスが可能なSRAM等の半導体メモリで構成されている。

【0013】パージ・フリーズレジスタ43は、CPUの指示に従って、データメモリ52に記憶されたデータを無効化(パージ)するか書換え禁止(フリーズ)にするかを設定するとともに、CPUが指定する優先順位を設定する。LRUパージ・フリーズ制御部42は、パージ・フリーズレジスタ43に設定されたデータを用いて、LRUアルゴリズムに基づきタグメモリ51のバリッドビットの書換えを行なう。

【0014】LRUメモリ優先順位情報更新制御部44は、LRUパージ・フリーズ制御部42の制御により、LRUメモリ40に記憶された優先順位情報の書換えを行なう。LRUメモリ40は、優先順位情報を記憶する。アドレスコンパレータ57は、CPUやDMAコントローラから与えられるアドレスと、このアドレスに従ってタグメモリ51から読み出されたアドレスデータとを比較し、キャッシュヒットか否かを判定する。

【0015】制御回路58は、アドレスコンパレータ57の比較結果、LRUメモリ40から出力された優先順位情報、及び、外部からのタイミング制御用のリードイネーブル信号等に応答して、キャッシュメモリ52やセクタ59の入出力を制御する。セクタ59は、外部からのデータをキャッシュメモリ52のどのウェイに入力するか、又はどのウェイのデータを出力するかを切り換える。

【0016】次に、上記公報に記載の半導体記憶装置の動作について説明する。CPU、DMAコントローラ等によりアドレスが外部から与えられると、このアドレスは、タグメモリ51に与えられ、ここでデータメモリ52のアドレスに変換されるとともに、アドレスコンパレータ57に入力される。

【0017】また、各データの優先順位情報は、LRU

メモリ40に記憶される。アドレスコンパレータ57は、外部から入力されたアドレスとタグメモリ51のアドレスとを比較し、CPUのリード動作においてキャッシュヒットかミスヒットかを判定する。キャッシュヒットであれば、データメモリ52のデータを外部のCPUに出力し、ミスヒットであれば、主記憶装置（図示せず）の記憶内容をデータメモリ52に読み込む。

【0018】また、CPUからのパージ・フリーズ命令及びCPUの命令に応じて指定する優先順位は、パージ・フリーズレジスタ43に書き込まれる。LRUパージ・フリーズ制御部42は、優先順位の情報に従って、LRUメモリ優先順位情報更新制御部44を制御することにより、LRUメモリ40内の優先順位情報を操作し、最適なパージ・フリーズ状態を自動的に設定する。パージ処理の場合、LRUパージ・フリーズ制御部42は、タグメモリ51内にあるバリッドビットを同時に操作し、タグアドレスに対応するデータを無効にする。

【0019】図14は、LRUパージ・フリーズ制御部42のフリーズ処理を示す流れ図である。まず、フリーズの対象となる優先順位をパージ・フリーズレジスタ43から読み出す（ステップS1）。次に、図示しないLRUメモリの全エントリから順に優先順位情報を読み出し（ステップS2）、該当する優先順位のウェイが必ず他のウェイよりも高い優先順位となるように設定する（ステップS3、S4）。

【0020】ただし、その際、優先順位が高くなるように毎回設定するのではなく、優先順位情報を読み出すときにのみ疑似的に優先順位が高いかのようにみせる処理を施す。例えば、ウェイAにフリーズしておきたいデータがあるとき、ウェイDがヒットしたとしてもウェイAの優先順位を疑似的に高めておくことにより、ウェイAのデータにプロテクトをかけることができる。これは、使用しないデータは優先順位が自動的に下がるというハードウェア上の制約があるからである。

【0021】一方、LRUパージ・フリーズ制御部42のパージ処理では、まず、パージの対象となる優先順位をパージ・フリーズレジスタ43から読み出す。次に、LRUメモリの全エントリから優先順位情報を順に読み出し、該当する優先順位のウェイが最下位の優先順位となるように優先順位情報を操作し更新する。更に、タグメモリ51内にあるバリッドビット（有効ビット）をインバリッドにし、該当するデータを無効にする。

【0022】以上のような記憶制御手順を実行することにより、図15における斜線を付した部分、即ち、この例では、ウェイ52A、52B、52Cのうちの優先順位3に該当するエントリ52Bb、52Bc、52Ca、…のみをパージ／フリーズすることができ、上述した頻繁に参照されるデータ群のようなデータのみをパージ／フリーズすることが可能となる。図15は、従来の各記憶データの優先順位を考慮したパージ／フリーズ制

御の概要を示す図である。

【0023】このように、上記公報に記載の半導体記憶装置では、LRU制御によってフリーズ命令を考慮しており、パージ／フリーズ命令と指定する優先順位とがパージ・フリーズレジスタ43に書き込まれると、LRUパージ／フリーズ制御部42が、LRUメモリ優先順位情報の更新制御部44を制御することによって、LRUメモリ40内の優先順位情報を操作し、所定のパージ／フリーズ状態を設定する。

【0024】

【発明が解決しようとする課題】上記従来の半導体記憶装置では、優先順位情報を読み出すときにのみ所定の処理を施せばよいものの、その都度に、LRUメモリ40内の全エントリに対して優先順位を書き換える処理を行わなければならない。また、CPUがキャッシュメモリの所定のアドレスをアクセスしてミスヒットした場合、キャッシュメモリは、LRUメモリ40を参照して、最も古くアクセスされたウェイを選択し、この選択したウェイのデータメモリ52に新しいデータを書き込むとともに、LRUメモリ40の記憶内容を書き換える。

【0025】図15に示すように、LRUメモリ40は、エントリ40a、40b、40c…を有している。また、データメモリ52のウェイ52Aはエントリ52Aa～52Ac、ウェイ52Bはエントリ52Ba～52Bc、ウェイ52Cはエントリ52Ca～52Ccを夫々有している。例えば、エントリ40bのウェイB（WAY-B）をフリーズして、LRUメモリ40をC>A>Bとなるように書き換えたとする。この場合、Cが最も最近にアクセスされたものであり、Bが最も古くアクセスされたことを示す。この後に、ミスヒットが生じた場合、WAY-Bに新しいデータが書き込まれることになり、優先度の高いWAY-Cが書き換えられることはない。このとき、WAY-Bが直近にアクセスされたことになるので、キャッシュメモリはLRUメモリ40を書き換えて、B>C>Aとする。

【0026】このように、上述のフリーズ処理を行っても、ミスヒットするたびにLRUメモリ40の内容が書き換えられるので、ミスヒットのたびにフリーズ処理を実行しなければ、ウェイ数に相当する回数のミスヒットの発生により、フリーズ処理したはずの記憶内容が消えることになる。逆に、ミスヒットするたびにフリーズ処理を実行することになると、エントリ数に相当する回数のフリーズ処理を実行しなければならないので、キャッシュメモリに余分な負荷がかかり、キャッシュメモリのスループットが低下するという問題を引き起こす。

【0027】上記問題を回避するため、前述の公報に記載の半導体記憶装置では、LRUメモリ40に記憶するウェイのアクセス履歴の数を（総ウェイ数-1）に制限している。例えば4ウェイのときには、3ウェイ分だけのアクセス履歴を記憶するように構成されている。この

ため、アクセス履歴が最も古くなっても、3番目より古くなることはないので、フリーズ処理したウエイが他のデータで置き換えられることはない。しかし、これは、ミスヒットした場合、残りの1ウエイのみを使って書き換えることを意味しており、複数のウエイを設けた本来の価値が発揮できない。

【0028】例えば、図15では、WAY-A~WAY-Cまでのアクセス履歴をLRUメモリ40に記憶し、残りのWAY-Dのアクセス履歴については何ら考慮しないようになっている。従って、WAY-A~WAY-Cのいずれかのウエイがヒットした場合には、アクセス履歴を書き換えるが、WAY-Dがヒットした場合の処理については何ら考慮されていない。仮に、WAY-A~WAY-Dのいずれにもヒットしなかった（ミスヒットした）場合、上記従来の半導体記憶装置では、上述の理由により、WAY-A~WAY-Cを書き換えることができないので、WAY-Dに新しいデータを取り込むと推察される。言い換えれば、4ウエイのキャッシュメモリでありながら、1ウエイのキャッシュメモリとしてのみ機能することになる。これは、データのヒット確率を低下させるとともに、CPUからみたデータのスループットを著しく低下させることになる。

【0029】更に、複数ウエイのうち1ウエイのみのフリーズ処理にしか対応できないので、複数ウエイにまたがるような中規模のプログラムをキャッシュメモリに取り込みたい場合、中規模のプログラム複数の一部分のみフリーズ処理することになる。このため、残りのプログラム部分は、必要ときに再度取り込まなければならず、十分なフリーズ機能を果たせない。

【0030】本発明は、上記に鑑み、簡素な回路構成を備えながらも、キャッシュメモリのスループットを低下させることなく2以上のウエイを対象に円滑なフリーズ処理が実行でき、ニーズに対応したキャッシュメモリ機能が実現できるキャッシュメモリ装置及びその制御方法を提供することを目的とする。

【0031】本発明は更に、上記目的を達成した上で、複数のウエイを対象としたフリーズ処理を可能とし、比較的規模が大きいデータでも格納が可能となるキャッシュメモリ装置を提供することを目的とする。

【0032】

【課題を解決するための手段】上記目的を達成するためには、本発明のキャッシュメモリ装置は、LRUデコード方式を採用し、Nを2以上の整数としたNウエイ・セット・アソシアティブ方式のキャッシュメモリ装置において、キャッシュメモリ各ウエイのアクセス履歴情報を記憶するLRUメモリと、各ウエイのフリーズ情報を記憶するフリーズメモリと、記憶されたアクセス履歴情報を修正することなく該アクセス履歴情報と前記フリーズ情報とに基づいて、特定のウエイをフリーズするためのリプレース信号を生成するリプレース判定手段とを備え

ることを特徴とする。

【0033】本発明のキャッシュメモリ装置では、アクセス履歴情報の書込み後にLRUメモリから読み出したアクセス履歴情報とフリーズメモリから読み出したフリーズ情報とに基づいてリプレース信号を生成する。このため、従来の半導体記憶装置の場合のようにアクセス履歴情報（ライトデータ）の書換えの度にページ・フリーズレジスタを参照してヒット情報を加味しつつ行うような処理が不要となり、複数種類のフリーズ情報を適宜選択することで、複数ウエイに対応するリプレース信号が容易に生成できる。これにより、キャッシュメモリのスループットを低下させず2以上のウエイを対象とした円滑なフリーズ処理が可能になり、ニーズに応じたキャッシュメモリ機能を簡素な回路構成で得ることが可能となる。また、LRUの元データを破棄することがないので、フリーズ状態を解除したとき、それまでのアクセス履歴に基づいた通常のLRU方式に直ちに復帰できる。

【0034】ここで、前記リプレース判定手段が、前記フリーズ情報と前記アクセス履歴情報とを取り込んで中間情報を生成するフリーズ制御回路と、前記フリーズ制御回路からの前記中間情報に対応して前記リプレース信号を生成するLRUデコード回路とを備えることが好ましい。この場合、リプレース判定手段を簡素な回路構成で実現することができる。

【0035】また、前記フリーズ情報としてN-1ウエイに対応する記憶容量を備えることが好ましい。この場合、Nウエイの内の1ウエイのみをリプレース対象として残し、他のウエイを全てフリーズ対象とすることができるので、従来に比して書換え回数が大幅に減少する。例えば、1つのサブルーチンを1ウエイ相当のメモリに格納しようとしてもデータ量が多くて格納できないような場合でも、N-1のウエイの活用が実現することにより、複数のウエイを対象としたフリーズ処理が可能となり、比較的規模が大きいデータでも格納することが可能となる。

【0036】更に好ましくは、前記リプレース信号が、各ウエイに夫々対応して出力されるビット値から成る。この場合、LRUデコード回路を簡素な回路構成から得ることができる。

【0037】本発明のキャッシュメモリ装置の制御方法は、LRUデコード方式を採用し、Nを2以上の整数としたNウエイ・セット・アソシアティブ方式のキャッシュメモリ装置を制御する制御方法において、キャッシュメモリ各ウエイのアクセス履歴情報を記憶し、キャッシュメモリ各ウエイのフリーズ情報を記憶し、記憶された前記アクセス履歴情報を修正することなく該アクセス履歴情報と前記フリーズ情報とに基づいてリプレース対象のウエイを算出することを特徴とする。

【0038】本発明のキャッシュメモリ装置の制御方法では、アクセス履歴情報の書込み後に読み出したアクセ

ス履歴情報とフリーズ情報とに基づいてリブレース対象のウエイを算出するので、複数種類のフリーズ情報を適宜選択することで、複数ウエイに対応するリブレース信号を容易に生成することができ、ニーズに応じたキャッシュメモリ機能が実現できる。また、LRUの元データを破棄することがないので、フリーズ状態を解除したとき、それまでのアクセス履歴に基づいた通常のLRU方式に直ちに復帰できる。

【0039】ここで、本発明では、前記アクセス履歴情報と前記フリーズ情報とに基づいてリブレース対象のウエイを算出する際に、前記フリーズ情報に対応して前記アクセス履歴情報を中間情報に変換した後にリブレース対象のウエイを算出することが好ましい。この場合、既に設計済みのキャッシュメモリ装置の資産を継承しながら、簡単な論理回路を付加するだけで、フリーズ機能付きのキャッシュメモリ装置を実現できる。また、LRUの元データを破棄することがないので、フリーズ状態を解除したとき、それまでのアクセス履歴に基づいた通常のLRU方式に即座に復帰できる。

【0040】更に好ましくは、前記アクセス履歴情報と前記フリーズ情報とに基づいてリブレース対象のウエイを算出する際に、Nウエイのアクセス順序を最近アクセ

$$\text{OUT}[X] = \prod_{j=0}^{X-1} M[j, X] \times \prod_{j=X+1}^{N-1} M[X, j]$$

(但し、 $0 \leq X, j \leq (N-1)$ 、 $[,]$  内の数字で左側の数<右側の数)

に基づいて行われる。この場合、上記中間情報を生成することにより、従来のリブレース対象ウエイの算出回路をそのまま利用することも可能となる。

【0043】

【発明の実施の形態】図面を参照して本発明を更に詳細に説明する。図1は、本発明の一実施形態例におけるコンピュータシステムの構成を示すブロック図である。

【0044】本実施形態例におけるコンピュータシステムは、CPU10と、コピーバック方式を採用したキャッシュメモリ装置11と、バスコントロールユニット(以下、BCUと呼ぶ)12と、主記憶装置14と、システムバス13とから概略構成されている。ここでは、LRUアルゴリズムに基づく4ウエイ・セット・アソシアティブ方式のキャッシュメモリ装置11を例に説明するが、本発明は4ウエイに限定されるものではない。

【0045】CPU10は、コンピュータシステムの各部を制御してデータ処理を行うと共に、キャッシュメモリ装置11に対し、所定のアドレス信号ADを供給して、所望の命令やデータ等(以下、記憶データと呼ぶ)の授受を行う。

【0046】キャッシュメモリ装置11は、要求されたアドレス信号ADに対応する記憶データDTM[]が自己装置内に存在するとき(以下、ヒットと呼ぶ)、CP

スされたウエイから順に並べたとき、フリーズ対象のウエイが1乃至N-1のいずれかのアクセス順序になるような中間情報を生成した後にリブレース対象のウエイを算出する。この場合、LRUの機能を損なうことなく、リブレース対象のウエイを算出することができる。

【0041】具体的には、ウエイAに対するウエイBのアクセス順序を表すアクセス履歴情報をW[A, B]とし、ウエイA及びウエイBのフリーズ情報を夫々FRZ[A]及びFRZ[B]とすると、前記中間情報としてのM[A, B]への変換が、

$$M[A, B] = W[A, B] + \text{FRZ}[A] \quad (\text{但し、} 0 \leq A < B = (N-1))$$

$$M[A, B] = (W[A, B] + \text{FRZ}[A]) \times !\text{FRZ}[B]$$

(但し、 $0 \leq A < B < (N-1)$ であり、! $\text{FRZ}[]$ は $\text{FRZ}[]$ の反転値を示す)に基づいて行われる。この場合、簡単な論理で、フリーズ対象のウエイがリブレース対象のウエイにならないように変換処理することができるので、小規模な論理回路を追加するだけでフリーズ機能付きのキャッシュメモリ装置が実現できる。

【0042】具体的には、リブレース対象のウエイをXとすると、該ウエイXの算出は、次式

【数2】

U10に対して記憶データDTを供給する。

【0047】一方、要求されたアドレス信号ADに対応する記憶データDTM[]が自己装置内に存在しないとき(以下、ミスヒットと呼ぶ)、キャッシュメモリ装置11は、当該記憶データDTM[]を主記憶装置14から取得するために、BCU12に対しアドレス信号BADを送ってメモリリードサイクルの発行を要求する。ここで、各信号名の後の[]内にはウエイの番号が入るが、その記載を省略している。

【0048】BCU12は、システムバス13を介してアドレス信号BADを主記憶装置14に送り、主記憶装置14から記憶データBDTを取得する。その後、BCU12は、キャッシュメモリ装置11に記憶データBDTを供給し、キャッシュメモリ装置11は、記憶データBDTを、所定の場所に記憶するとともにCPU10に対して供給する。同時に、キャッシュメモリ装置11は、読み込んだ記憶データBDTを自装置内に記憶する。このとき、自装置内に書き込む領域が空いていないときには、前回のアクセス履歴が最も古いウエイの記憶データを主記憶装置14に書き戻し(コピーバック)してから、新たに読み出した記憶データBDTで書き換える。

【0049】このようにして、CPU10は、所望のア

ドレスの記憶データをキャッシュメモリ装置11との間で授受することができる。

【0050】図1に示すように、キャッシュメモリ装置11は、タグメモリ部31、比較部32、データメモリ部33、セクタ部34、LRU部35、フリーズメモリ部36、リプレース判定回路37、及びコントローラ38から概略構成されている。

【0051】本実施形態例では、主記憶装置14の記憶容量を1MB（Bはバイトを意味する）であるとし、データメモリ部33の記憶容量を1kBであるとする。データメモリ部33は4個のウエイから成り、各ウエイは夫々64個のエントリに分割され、各エントリは4B（32ビット）の記憶データDTM[]を記憶している。1ウエイの記憶容量は256B（ $=64 \times 4$ ）であり、4ウエイでは1kBとなる。そして、主記憶装置14もこのようなデータメモリ部33の構成に対応して、256B毎の4096（ $1\text{MB}/256\text{B}$ ）個のブロックに分割されている。

【0052】各記憶データDTM[]をアクセスするためにCPU10から供給されるアドレス信号ADのビット長は20ビットであり、下位2ビット{1:0}は、4バイトのデータのうち1バイトを選択するためのバイトアドレスである。また、アドレス信号ADの中位6ビット{7:2}は、64個のエントリのいずれか1つを選択するためのインデックスアドレスIADであり、上位12ビット{19:8}は、主記憶装置14の4096個あるブロックのいずれか1つを選択するためのタグアドレスTADである。

【0053】タグメモリ部31は、各ウエイの各エントリに対応するタグアドレスTADと、主記憶装置14とデータメモリ部33との記憶内容が一致しているか否かを示すダーティビットDRTとを記憶している。タグメモリ部31は、コントローラ38が供給するアドレス信号ADTの中位6ビットのインデックスアドレスIADTに基づいてタグメモリ部31をアクセスし、そこに記憶されているタグアドレスTADを比較部32に出力する。ミスヒットして新たな記憶データDTM[]がデータメモリ部33に取り込まれたときには、タグメモリ部31は、新しいタグアドレスTADを所定のウエイと所定のエントリに対応した位置に記憶する。

【0054】また、データメモリ部33の記憶内容を更新する前に、コントローラ38はタグメモリ部31からダーティビットDRTO[]を読み取り、DRTO[]が“1”であれば、キャッシュメモリ内部のデータを追い出し、データメモリ部33の記憶データDTM[]を主記憶装置14にコピーバックする。ダーティビットDRTO[]が“0”であれば、コントローラ38は新たな記憶データDTM[]でデータメモリ部33を上書きする。

【0055】比較部32はウエイ数分の比較回路を有す

る。各比較回路は、コントローラ38から出力されるアドレス信号ADTの上位12ビットのタグアドレスTADと、タグメモリ部31から出力されるタグアドレスTAM[]とを比較して、一致した場合にはヒット信号（HIT）として“1”を出力し、不一致の場合にはミスヒット信号として“0”を出力する。ここで、各ウエイ0～3に対応したヒット信号を夫々HIT[0]～HIT[3]とする。

【0056】データメモリ部33は、64エントリ×4ウエイ分の記憶データDTM[]を有しており、インデックスアドレスIADDが入力されると、そのアドレスに対応した4ウエイ分の記憶データDTM[]を出力又は入力する。

【0057】セクタ部34は、ヒット信号HIT[]に対応した選択信号DSL[]により、データメモリ部33から読み出された記憶データDTM[]のうちでヒットしたウエイの記憶データDTM[]を選択し、DTD[]としてコントローラ38に出力し、或いは、リプレース信号OUT[]に対応した選択信号DSL[]により、コントローラ38から供給される記憶データDTD[]を置き換えるウエイを選択してデータメモリ部33に出力する。

【0058】LRUメモリ部35は、データメモリ部33内の各ウエイにアクセスした順序である履歴情報を各エントリ毎に記憶しており、コントローラ38から供給されるインデックスアドレスIADLが入力されると、そのアドレスに対応した6ビットの履歴情報W[]を出力する。キャッシュにヒットした場合、比較部32からヒット信号HIT[]が出力されるたびに、コントローラ38からヒット信号HIT[]に対応した新規アクセスウエイ信号NAW[]がLRUメモリ部35に供給される。これに基づいてLRUメモリ部35は、新たな履歴情報としてLRUメモリ内の記憶内容を更新する。このときは、 $\text{NAW}[n] = \text{HIT}[n]$ （ $n=0 \sim 3$ ）である。

【0059】一方、ミスヒットしたとき、LRUメモリ部35は、リプレース信号OUT[]に対応した新規アクセスウエイ信号NAW[]がコントローラ38から供給される。これに基づいてLRUメモリ部35は、LRUメモリからインデックスアドレスIADTに対応する履歴情報を読み出し、最も過去にアクセスされたウエイの情報DSL[]をセクタ部34に出力し、そのウエイのデータメモリ部33に新たな記憶データDTM[]を書き込む。このときは、 $\text{NAW}[n] = \text{OUT}[n]$ （ $n=0 \sim 3$ ）である。

【0060】フリーズメモリ部36は、各エントリ毎に該当ウエイのフリーズの有無を記憶しており、コントローラ38から供給されるインデックスアドレスIADLが入力されると、そのアドレスに対応した3ビットのフリーズ情報FRZ[]を出力する。また、フリーズメモ

リ部36は、コントローラ38からフリーズ書込データ(FRZライトデータ)FW[]とフリーズ書込許可信号(FRZライトイネーブル信号)FE[]とが入力されると、インデックスアドレスIADLに対応した位置にフリーズ情報FRZ[]を書き込む。3ビットのFRZ[]は、4つのウエイのうち任意の3ウエイに対応させることができる。以下の説明では、ウエイ0~2をフリーズ可能なウエイとして、これらのフリーズ情報をFRZ[0]~FRZ[2]に夫々対応させている。

【0061】リプレース判定部37は、LRUメモリ部35から出力される履歴情報W[]と、フリーズメモリ部36から出力されるフリーズ情報FRZ[]とに基づいて、どのウエイを新たな情報で置き換えるかを表すリプレース信号OUT[]を出力する。リプレース信号OUT[]は、各ウエイ0~3に対応してOUT[0]~OUT[3]がある。

【0062】次に、キャッシュメモリ装置11の全体的な動作を図1を参照して説明する。

【0063】[キャッシュ・ヒット動作]CPU10から供給されたアドレス信号ADは、一旦、コントローラ38に保持される。アドレス信号ADのうち中位6ビットのインデックスアドレスIADD及びIADTが夫々、データメモリ部33及びタグメモリ部31に供給される。この場合、インデックスアドレスIADDとIADTとは同一の値である。このインデックスアドレスIADTに対応するタグメモリ部31の内容が各ウエイ0~3から読み出され、比較部32内の4つの比較回路に夫々入力される。これと同時に、インデックスアドレスIADDに対応する4ウエイ分の記憶データDTM[]がセクタ部34に読み出される。

【0064】一方、コントローラ38から出力されたアドレス信号ADTのうち、タグアドレスTADは、比較部32のもう一方の入力端子に入力され、タグメモリ31から読み出されたタグアドレスTAM[]と比較される。比較部32は、4ウエイのうちいずれか1つのウエイが一致したことを示す信号HIT[]="1"を出力する。その他のウエイに対応するヒット信号HIT[]は"0"(ミスヒット)となる。

【0065】コントローラ38は、上記HIT[]に基づいて選択信号DSL[]と新規アクセスウエイ信号NAW[]とを生成し、選択信号DSL[]をセクタ部34に、新規アクセスウエイ信号NAW[]をLRUメモリ部35に夫々供給する。セクタ部34は、選択信号DSL[]に対応するウエイの記憶データDTM[]を選択し、CPU10に対して出力する。

【0066】その後、LRUメモリ部35は、ヒット信号HIT[]に対応する新規アクセスウエイ信号NAW[]に基づいて、ヒットしたウエイのアクセスが最近されたかのように履歴情報を書き換える。このとき、NAW[]=HIT[]である。

【0067】[キャッシュ・ミスヒット動作]次に、ミスヒットした場合の動作を説明する。比較処理まではヒット時と同じである。

【0068】比較部32が、4ウエイのうちいずれのウエイも一致しなかったことを示す信号HIT[0]~HIT[3]="0"を出力すると、コントローラ38は保持していたアドレス信号BADをBCU12に送り、BCU12はそのアドレスBADに対応する記憶データBDTを主記憶装置14から読み出す。

【0069】コントローラ38は、記憶データBDTの読出しと同時に、アドレス信号BADに対応したインデックスアドレスIADLをLRUメモリ部35とフリーズメモリ部36とに出力する。これに回答して、LRUメモリ部35が履歴情報W[]を、フリーズメモリ部36がフリーズ情報FRZ[]をリプレース判定部37に夫々出力する。リプレース判定部37は、履歴情報W[]とフリーズ情報FRZ[]とに基づいてリプレース信号OUT[]を算出し、コントローラ38に出力する。これにより、コントローラ38は、リプレース信号OUT[]に対応した選択信号DSL[]をセクタ部34に供給する。

【0070】BCU12を介して読み出された記憶データBDTは、コントローラ38に入力され、CPU10に対して出力される。また、コントローラ38は、アドレス信号BADに対応するインデックスアドレスIADDをデータメモリ部33に供給するとともに、記憶データBDTをDTDとしてセクタ部34に供給する。このとき、コントローラ38は、タグメモリ部31に含まれるダーティビット情報DRT0[]を確認し、データメモリ部33の記憶内容が主記憶装置14の記憶内容と異なるときには、データメモリ部33の記憶内容を主記憶装置14に書き戻す。その後、セクタ部34が、データメモリ部33内の選択信号DSL[]で指定されたウエイに記憶データDTM[] (=DTD)を出力する。データメモリ部33は、インデックスアドレスIADDで指定された位置に記憶データDTM[]を記憶する。

【0071】次いで、コントローラ38は、書き換えたウエイの情報である新規アクセスウエイ信号NAW[]をLRUメモリ部35に出力する。LRUメモリ部35は、NAW[]に基づいて、書き換えたウエイのアクセスが最近されたかのように履歴情報を書き換える。ここでは、NAW[n]=OUT[n](n=0~3)である。

【0072】以上は、キャッシュメモリ装置11から記憶データを読み出す場合を説明したが、書き込む場合も、ほぼ同様に行われる。

【0073】次に、LRUメモリ部35の基礎技術について詳細に説明する。まず、LRUメモリが記憶している履歴情報について説明する。図2は、Nウエイ分のア



クセス履歴を示す模式図であり、(a)はN=2の場合、(b)はN=3の場合、(c)はN=4の場合、(d)はN=5の場合を夫々示す。各図において、矢印の基点側に位置するウエイが矢印の終端側に位置するウエイよりも最近にアクセスされたとき、LRUメモリに“0”を記憶し、アクセスの順序関係が逆の場合には“1”を記憶する。

【0074】図2(a)における2ウエイでは、LRUメモリは、1ビットの履歴情報×エントリ数のメモリで構成される。例えば、履歴情報W[0,1]に“0”が記憶されているとき、LRUメモリ部35は、ウエイ1がウエイ0よりも最新アクセスされたと判定し、逆に、履歴情報W[0,1]に“1”が記憶されているとき、ウエイ0がウエイ1よりも最新アクセスされたと判定する。

【0075】また、履歴情報W[0,1]に“0”が記憶されているときに、ウエイ0がヒットすると、履歴情報W[0,1]は“1”に書き換えられ、ウエイ1がヒットすると、履歴情報W[0,1]は“0”のままである。逆に、履歴情報W[0,1]に“1”が記憶されているときに、ウエイ1がヒットすると、履歴情報W[0,1]は“0”に書き換えられ、ウエイ0がヒットすると、履歴情報W[0,1]は“1”のままである。

【0076】図2(b)における3ウエイでは、LRUメモリは、3ビットの履歴情報W[0,1]、W[0,2]、W[1,2]を記憶している。例えば、各ウエイが「2←1←0」の順にアクセスされたとき、LRUメモリは、履歴情報(W[0,1]、W[0,2]、W[1,2])として、(0,0,0)を記憶している。即ち、ウエイ0よりもウエイ1の方が最近アクセスされているのでW[0,1]は“0”であり、ウエイ0よりもウエイ2の方が最近アクセスされているのでW[0,2]は“0”であり、ウエイ1よりもウエイ2の方が最近アクセスされているのでW

[1,2]は“0”である。

【0077】この状態でウエイ1がヒットすると、アクセス履歴は「1←2←0」となり、履歴情報(W[0,1]、W[0,2]、W[1,2])は、(0,0,1)に書き換えられる。即ち、ウエイ2よりウエイ1の方が最近アクセスされたことになるのでW[1,2]は“1”になる。なお、3ウエイのLRUメモリの容量は、3ビットの履歴情報×エントリ数である。

【0078】図2(c)における4ウエイ・セットアソシエティブ・キャッシュのLRU方式では、LRUメモリ部35内のLRUメモリは、図3に示すように、6ビットの履歴情報W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3]でアクセス履歴を記憶している。図3は、4ウエイの全てのアクセス順序と履歴情報の関係を、ヒットしたウエイを基準に示す履歴情報図である。

【0079】例えば、各ウエイが「2←1←3←0」の順にアクセスされたとき、LRUメモリは、履歴情報(W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3])として、(0,0,0,0,1,1)を記憶している。この状態でウエイ1がヒットすると、アクセス履歴は「1←2←3←0」となり、履歴情報(W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3])は、(0,0,0,1,1,1)に書き換えられる。なお、4ウエイのLRUメモリの容量は、6ビットの履歴情報×エントリ数である。図2(d)における5ウエイも、同様にして10ビットの履歴情報で記憶している。

【0080】図2(a)～(d)の各場合において履歴情報として夫々必要なビット数は、ウエイ数をNとしたとき、次式

【数3】

$$\text{履歴情報のビット数} = \sum_{i=1}^N (i-1) = N(N-1)/2$$

で表され、N=2のときに1ビットとなり、N=3のときに1+2=3ビットとなり、N=4のときに1+2+3=6ビットとなり、N=5のときに1+2+3+4=10ビットとなる。

【0081】図5は、図1に示したタグメモリ部31と比較部32との詳細な構成を示すブロック図である。タグメモリ部31は、TAGアドレスデコーダ61、TAG書込み制御回路62、及び、4ウエイ分のタグメモリ31a～31dで構成され、比較部32は、4つの比較回路32a～32dで構成される。

【0082】タグメモリ31a～31dは、各ウエイの各エントリに対応するタグアドレスTADと、主記憶装置14とデータメモリ部33との記憶内容が一致しているか否かを示すダーティビットDRTとを記憶している。タグメモリ31a～31dは、ウエイ数×エントリ数×(アドレス線数+1ビット)分の記憶容量を有し、

本実施形態例では、4ウエイ×64エントリ×(14ビット+1ビット)=3840ビットの記憶容量のSRAM(Static Random Access Memory)で構成される。

【0083】TAGアドレスデコーダ61は、コントローラ38(図1)から供給されるアドレス信号ADTの中位6ビットであるインデックスアドレスIADTが入力され、これをデコードしてタグメモリ31a～31dのワード線(エントリに相当)の1つを活性化する。タグメモリ31a～31dは、選択されたエントリのTAG情報TAM[0]～TAM[3]を比較部32に出力するとともに、ダーティビットDRT0[]をコントローラ38に出力する。

【0084】TAG書込み制御回路62は、コントローラ38からインデックスアドレスIADT、選択信号DSL[]、及び、ダーティビット書込み情報DRTI[]が入力される。ミスヒットしたときに、TAG書込み制

御回路62は、選択信号DSL[]で指定されるウエイのインデックスアドレスIADTで指定されるエントリに、新たなタグアドレスTADを書き込むとともに、ダーティビットDRTに“0”を書き込む。ヒットしたときには、TAG書き込み制御回路62は、タグアドレスTADはもとのままで、ダーティビットDRTに“1”を書き込む。

【0085】タグメモリ31a~31dは、コントローラ38(図1)が供給するアドレス信号ADTの中位6ビットのインデックスアドレスIADTで指定されるタグアドレスTAM[0]~TAM[3]を比較回路32a~32dに夫々出力し、TAG書き込み制御回路62から供給されるタグアドレスTADを記憶する。同様に、ダーティビットDRTを読み書きする。読み出されたダーティビットDRTは、ダーティビット読出情報DRT0[]として上記のようにコントローラ38に出力される。

【0086】コントローラ38は、読み出したダーティビット読出情報DRT0[]を確認し、DRT0[]が“1”であれば、データメモリ部33の内容を更新する前に、データメモリ部33(図1)の記憶データDTDを主記憶装置14にコピーバックする。DRT0[]が“0”であれば、コントローラ38は新たな記憶データDTDでデータメモリ部33を上書きする。コントローラ38は更に、上記機能に加え、CPU10が出力するアドレス信号ADTを保持して、アドレスデータの上位ビットであるタグアドレスTADと、中位ビットであるインデックスアドレスIADと、下位ビットであるバイトアドレス(ワードアドレス)とを出力する。

【0087】図6は、図1に示したデータメモリ部33の詳細な構成を示すブロック図である。データメモリ部33は、DM(データメモリ)アドレスデコーダ64と、4ウエイ分のデータメモリ33a~33dとで構成され、タグメモリ31に記憶されたタグアドレスTADと、インデックスアドレスIADに対応する主記憶装置14の記憶データDTとを記憶している。

【0088】データメモリ33a~33dは、タグメモリ31a~31dに記憶されたタグアドレスTADとインデックスアドレスIADDとで指定される主記憶装置14の記憶データDTDを記憶している。このインデックスアドレスIADDは、前述のインデックスアドレスIADLと、アクセスのタイミングの点で異なる。データメモリ33a~33dは、ウエイ数×エントリ数×バイト数分の記憶容量を有し、本実施形態例では、4ウエイ×64エントリ×4バイト=1kBの記憶容量のSRAMで構成される。

【0089】DMアドレスデコーダ64は、コントローラ38から供給されるアドレス信号の中位6ビットであるインデックスアドレスIADDが入力され、これをデコードしてデータメモリ33a~33dのワード線(エ

ントリに相当)の1つを活性化する。データメモリ33a~33dは選択されたエントリの記憶データDTM[0]~DTM[3]をセクタ部34に出力する。

【0090】セクタ部34は、インデックスアドレスで指定される位置のデータメモリ部33から読み出した4ウエイ分の記憶データDTM[]のうち、比較回路32a~32dで一致が検出されたウエイに対応する記憶データDTM[]を1つ選択し、記憶データDTDとしてコントローラ38に出力する。ミスヒットしてデータメモリ部33の内容を新しい記憶データに書き換えるときも、同様に書き込み位置を選択して、主記憶装置14から読み出した記憶データDTDを書き込む。

【0091】つまり、セクタ部34は、データメモリ33a~33dから記憶データDTM[0]~DTM[3]が入力され、コントローラ38から選択信号DSL[]が入力された際にヒットすると、選択信号DSL[]で指定されるウエイの記憶データDTDを1つ選択してコントローラ38に出力する。ミスヒットしたときには、セクタ部34は、主記憶装置14から読み込んだ記憶データBDTを、選択信号DSL[]で指定されるウエイのインデックスアドレスIADDで指定されるエントリに、新たな記憶データDTM[]として書き込む。

【0092】コントローラ38はセクタ65を内蔵している。セクタ65は、ヒット信号HIT[]又はリプレース信号OUT[]のいずれか一方を選択し、選択信号DSL[]としてセクタ部34に出力する。ヒットしたときは、DSL[n]=HIT[n](n=0~3)であり、ミスヒットしたときは、DSL[n]=OUT[n](n=0~3)である。

【0093】図7は、図1に示したLRUメモリ部35の詳細なブロック図である。LRUメモリ部35は、LRUメモリ15、LRU書込データ生成回路16、LRUアドレスデコーダ66、及びLRU書き込み制御回路67で構成されている。

【0094】LRUメモリ15は、各エントリ毎に、キャッシュメモリの各ウエイのアクセスした順序を6ビットの履歴情報W[]で記憶している。各ウエイのアクセス順序と履歴情報W[]との関係は、図3及び図4に示す通りである。図4は、図3のアクセス履歴の表示順を並べ換えて、最も過去にアクセスされたウエイがまとまるようにしたものである。

【0095】図7に示すように、LRUメモリ15は、エントリ数×履歴情報数分の記憶容量を有しており、本実施形態例では、64エントリ数×6ビット=384ビットの記憶容量のSRAMで構成される。

【0096】LRUアドレスデコーダ66は、コントローラ38から供給されるアドレス信号の中位6ビットであるインデックスアドレスIADLが入力され、これをデコードしてLRUメモリ15のワード線(エントリに

相当)の1つを活性化する。LRUメモリ15は、選択されたエントリの履歴情報W[0,1]～W[2,3]をリプレース判定部37に出力する。

【0097】コントローラ38内のセクタ68は、ヒット信号HIT[]又はリプレース信号OUT[]のいずれか一方を選択し、新規アクセスウエイ情報NAW[]としてLRU書込データ生成回路16に出力する。ヒットしたときは、NAW[n]=HIT[n] (n=0～3)であり、ミスヒットしたときは、NAW[n]=OUT[n] (n=0～3)である。

【0098】LRU書込みデータ生成回路16は、新規アクセスウエイ情報NAW[]に基づいて、LRUメモリ15の内容を更新するために、6ビットのLRUライトデータLW[]と、LRUライトイネーブル信号LE[]とを生成し、LRU書込み制御回路67に出力して、LRUメモリ15への書込みを制御する。

【0099】いま、LRUメモリ15に対するLRUライトイネーブル信号をLE[A,B]と表記し、LRUライトデータをLW[A,B]と表記することにする。ウエイL (Lは $0 \leq L \leq 3$ の整数)がヒットした場合、LRUライトデータは、LW[A,L]=0、LW[L,B]=1 (但し $A < L < B$ )として定義し、このときLRUライトイネーブル信号を、LE[A,L]=LE[L,B]=1として定義する。LE[A,B]及びLW[A,B]におけるA,Bは夫々にウエイを示している。

【0100】LRUデコード論理では、「任意の2つの

ウエイA,Bのうち、どちらが最後に使用されたか」を全ウエイの組み合わせに対応して保持させる必要がある。一般に、Nウエイの場合には、 $N(N-1)/2$ ビットの情報で全ウエイのアクセス順を表現できることが知られている。このアクセス順は、LRUメモリ部35内のLRUメモリ15に記憶されており、アクセス順を表す6ビットの履歴情報W[0,1]～W[2,3]として保持されている。

【0101】この履歴情報W[]は、キャッシュメモリがアクセスされる毎に更新され、常に最新の履歴情報に書き換えられる。キャッシュにヒットしたときには、履歴情報は、ヒットしたウエイが一番最近にアクセスされた状態となるように書き換えられ、ミスヒットしたときには、新しいデータに置き換えられたウエイが一番最近にアクセスされた状態となるように書き換えられる。但し、最近ヒットしたデータに再ヒットしたときには書き換えなくても良い。

【0102】履歴情報を書き換えるとき、ヒット又は置き換えられたウエイの情報がLRU書込みデータ生成回路16に入力されると、LRU書込みデータ生成回路16は、表1に示すように、ヒットしたウエイに対応するLRUライトデータLW[A,B]を生成し、LRU書込み制御回路67を介してLRUメモリ15に出力する。

【0103】

【表1】

	LW[0,1]	LW[0,2]	LW[0,3]	LW[1,2]	LW[1,3]	LW[2,3]
ウエイ0ヒット	1	1	1	0	0	0
ウエイ1ヒット	0	0	0	1	1	0
ウエイ2ヒット	0	0	0	0	0	1
ウエイ3ヒット	0	0	0	0	0	0

【0104】LRU書込み制御回路16は、6ビットのLRUライトデータLW[A,B]をすべて書き込むわけではなく、所定の3ビットを書き換えるようにしている。表2に示すLRUライトイネーブル信号LE[A,B]によって、どのビットを書き換えるかを指定する。なお、LE[A,B]に対応するビット値の“0”は書込みをマスクすることを意味し、“1”は書込みを実行

することを意味する。つまり、LRU書込み制御回路16は、LRUライトイネーブル信号LE[A,B]が“1”であるLRUライトデータLW[A,B]をLRUメモリ15に書き込む。

【0105】

【表2】

	LE[0,1]	LE[0,2]	LE[0,3]	LE[1,2]	LE[1,3]	LE[2,3]
ウエイ0ヒット	1	1	1	0	0	0
ウエイ1ヒット	1	0	0	1	1	0
ウエイ2ヒット	0	1	0	1	0	1
ウエイ3ヒット	0	0	1	0	1	1

【0106】このように、ウエイLがヒットし又は書き換えられると、LRU書込みデータ生成回路16は、表1と表2とに従ってLW[A,B]とLE[A,B]と

を生成し、LRU書込み制御回路67は、LE[A,B]が“1”となっている位置に対応するLRUメモリ15の履歴情報W[A,B]のビット位置に、LW

[A, B] のビット値を書き込む。

【0107】例えば、ウエイ1がヒットした場合には、LRUライトイネーブル信号LE[0,1]、LE[1,2]、LE[1,3]が“1”であるので、LRU書き込み制御回路67は履歴情報W[0,1]、W[1,2]、W[1,3]にLRUライトデータ(LW[0,1]、LW[1,2]、LW[1,3])として(0, 1, 1)を書き込む。この結果、図3の破線で囲んだ領域に示すように、ウエイ1がヒットする前の状態がどのような履歴であっても、履歴情報W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3]は、(0, d, d, 1, 1, d)となる。ここで、dは“don't care”を意味し、更新する前のビット値が保持される。

【0108】図8は、図1に示したフリーズメモリ部36の詳細なブロック図を示す。フリーズメモリ部36は、フリーズメモリ20と、FRZアドレスデコーダ69と、FRZ書き込み制御回路70とで構成される。

【0109】フリーズメモリ20は、各エントリ毎に書き換え禁止にするウエイの情報を3ビットのフリーズ情報FRZ[]で記憶している。ウエイ0～2のフリーズ情報は夫々FRZ[0]～FRZ[2]に対応している。フリーズ情報FRZ[n]が“1”であるとき、そのウエイnは書き換えが禁止されていることを意味し、“0”であるとき、そのウエイnは新規の記憶データで書き換えが可能であることを意味する。

【0110】フリーズメモリ20は、(ウエイ数-1)×エントリ数分の記憶容量を有し、本実施形態例では、3ウエイ×64エントリ=192ビットの記憶容量のSRAM(Static Random Access Memory)で構成される。この記憶容量は、フリーズする領域に応じて適宜変更しても良い。例えば、4ウエイともフリーズするときには4×64ビットが必要であるし、また、1×64ビットでも良い。

【0111】FRZアドレスデコーダ69は、コントローラ38から供給されるアドレス信号の中位6ビットであるインデックスアドレスIADLが入力され、これをデコードしてフリーズメモリ20のワード線(エントリに相当)の1つを活性化する。フリーズメモリ20は、選択されたエントリのフリーズ情報FRZ[0]～FRZ[2]をリプレース判定部37に出力する。なお、FRZアドレスデコーダ69は、LRUアドレスデコーダ66と共用してもよく、或いは、LRUメモリ15とFRZメモリ20とを共通のワード線で接続したSRAMで構成してもよい。

【0112】FRZ書き込み制御回路70は、コントローラ38から供給されるFRZライトデータFW[]とFRZライトイネーブル信号FE[]とに基づいて、フリーズメモリ20内のインデックスアドレスIADLで指定される位置にフリーズ情報を書き込む。

【0113】いま、フリーズメモリ20に対するFRZ

ライトイネーブル信号をFE[A]と表記し、FRZライトデータをFW[A]と表記することにする。ウエイK(Kは $0 \leq K \leq 2$ の整数)をフリーズする場合、コントローラ38(図1)は、FRZライトデータFW[K]=1とし、FRZライトイネーブル信号をFE[K]=1、FE[A]=0(但し、 $A \neq K$ )としてFRZ書き込み制御回路70に出力する。FE[A]及びFW[A]におけるAはウエイを示している。

【0114】逆に、ウエイK(Kは $0 \leq K \leq 2$ の整数)のフリーズを解除する場合、コントローラ38は、FRZライトデータFW[K]=0、FRZライトイネーブル信号FE[K]=1をFRZ書き込み制御回路70に出力し、フリーズメモリ20に書き込む。

【0115】図9は、フリーズ情報をフリーズメモリ20に書き込む手順を示す流れ図である。同図に示す手順は、プログラム起動時に、主記憶装置14の所定の領域に記憶された記憶データを、フリーズI/O命令によってキャッシュメモリ装置11の所定のウエイにコピーしてフリーズする方法である。

【0116】また、図10は、主記憶装置14のアドレス空間の配置例を示す図である。アドレスADR0～(ADR4-1)はプログラム領域、ADR4以上はデータ領域とする。プログラム領域中のアドレスADR1とアドレスADR2～(ADR3-1)の領域に、フリーズしたいプログラムが記述されているとする。

【0117】データ領域中のアドレスADR5～(ADR6-1)の領域は、ウエイ0にフリーズしたいプログラムを転送するため、予め一時的にコピーしておくための作業領域である。同様に、アドレスADR6～(ADR7-1)、ADR7～(ADR8-1)領域は夫々、ウエイ1及びウエイ2用のデータ転送作業領域である。

【0118】次に、図1、及び図8～図10を参照して、フリーズメモリ20へのフリーズ情報の書き込み手順を説明する。ここでは、アドレスADR2～(ADR3-1)に存在するフリーズ対象プログラムをウエイ1に転送する場合を例に挙げて説明する。

【0119】まず、ステップS11で、アドレスADR2～(ADR3-1)に存在するフリーズ対象プログラムを、ウエイ1用データ転送作業領域ADR6～(ADR7-1)にブロック・コピーする。これは、予め組み込まれた標準のフリーズI/O命令によって、キャッシュメモリ装置11の所定のウエイにコピーするためである。

【0120】ステップS12で、CPU10は、フリーズI/O命令を実行し、フリーズ対象プログラムの開始/終了アドレスADR2/(ADR3-1)、コピー先のウエイ“1”をコントローラ38に設定する。ここで、CPU10は、データ転送作業領域ADR6～(ADR7-1)のアドレス情報をコントローラ38に別途供給してもよく、或いは、ウエイと1対1に対応してい

てウエイの情報“1”に基づいて生成できれば改めて設定しなくてもよい。

【0121】ステップS13で、コントローラ38は、BCU12にアドレスBADを出力し、フリーズ対象の記憶データBDTを主記憶装置14から読み込む。読出しアドレスBADはアドレスADR6から(ADR7-1)まで変化する。

【0122】ステップS14で、コントローラ38は、データメモリ部33にステップS13の読出しアドレスBADに対応するインデックスアドレスIADDを出力し、セクタ部34に記憶データBDTと選択信号DSL[1]=1とを出力することで、データメモリ部33のウエイ1に記憶データBDTを記憶する。

【0123】ステップS15で、コントローラ38は、LRUメモリ部35とフリーズメモリ部36とに、前記読出しアドレスBADに対応するインデックスアドレスIADLを出力し、LRUメモリ部35に新規アクセスウエイ情報NAW[1]=1を出力する。コントローラ38は更に、フリーズメモリ部36にFRZライトデータFW[1]=1、FRZライトイネーブル信号FE[1]=1を出力することで、フリーズメモリ20のアドレスIADLにFRZ[1]=1を書き込み、ウエイ1の該当アドレスをフリーズ状態に設定する。

【0124】ステップS16で、コントローラ38は、アドレスADRをインクリメントする。ステップS17で、コントローラ38は、アドレスADRが転送終了アドレスADR7未満であるかを判断し、ADR7未満であればステップS13～S16を繰り返し、ADR7以上であればフリーズ転送処理を終了する。

【0125】図11は、フリーズ情報をフリーズメモリ20に書き込むための更に別の手順を示す流れ図である。同図に示す手順は、プログラム実行中に、主記憶装置14の所定の領域に記憶された記憶データをフリーズ設定命令によってキャッシュメモリ装置11の所定のウエイにコピーしてフリーズする方法である。

【0126】次に、図1、図8、図10及び図11を参照して、フリーズメモリ20へのフリーズ情報の書き込み手順を説明する。ここでは、図10のアドレスADR1に存在するフリーズ対象プログラムをウエイ2に取り込み、フリーズ設定命令によってフリーズする例を説明する。

【0127】先ず、ステップS21において、CPU10(図1)は、主記憶装置14の所定のアドレスADR1に存在するプログラムを1行読み込む命令(以下、ロード命令とも呼ぶ)と、アドレスADR1とをコントローラ38に渡す。

【0128】ステップS22では、コントローラ38が、アドレスADR5に対応するインデックスアドレスIADTをタグメモリ部31に出力するとともに、タグアドレスTADを比較部32に出力する。

【0129】ステップS23において、コントローラ38は、比較部32がヒット信号HIT[]=1(ヒット)を出力すればステップS35の処理に進み、HIT[]=0(ミスヒット)であればステップS24の処理に進む。

【0130】ステップS24では、コントローラ38が、BCU12にBAD=ADR1を出力し、フリーズ対象記憶データBDTを主記憶装置14から読み込む。

【0131】ステップS25では、コントローラ38が、アドレスADR1に対応するインデックスアドレスIADLをフリーズメモリ部36に出力し、フリーズ情報FRZ[0]～FRZ[2]を取得する。

【0132】ステップS26では、コントローラ38が、フリーズ可能か否かを判断する。フリーズ情報FRZ[0]～FRZ[2]のいずれかが“0”であれば、フリーズ可能であるとしてステップS27に進む。フリーズ情報FRZ[0]～FRZ[2]が全て“1”であれば、フリーズ不可能として処理を終了する。

【0133】ステップS27では、コントローラ38が、アドレスADR1に対応するインデックスアドレスIADLをLRUメモリ部35に出力し、リブレース信号OUT[0]～OUT[3]を取得する。

【0134】ステップS28では、コントローラ38が、OUT[0]～OUT[2]のいずれかが“1”であればステップS30に進み、OUT[3]が“1”であればステップS29に進む。

【0135】ステップS29では、コントローラ38が、FRZ[2]→FRZ[1]→FRZ[0]の順にフリーズ情報を検証し、FRZ[F]が“0”でフリーズ対象となっていないウエイを探す。最初に検出したウエイ“F”をリブレース対象と決定し、ステップS31に進む。

【0136】ステップS30では、“1”となったOUT[F]に対応するウエイ“F”をリブレース対象と決定し、ステップS31に進む。

【0137】ステップS31では、コントローラ38が、アドレスADR1に対応するインデックスアドレスIADTをタグメモリ部31に出力し、ダーティビット読出情報DRT0[F]を取得する。

【0138】ステップS32では、コントローラ38が、ダーティビット読出情報DRT0[F]が“1”か否かを判断し、“1”であればステップS33に進み、“0”であればステップS34に進む。

【0139】ステップS33では、コントローラ38が、アドレスADR1に対応するインデックスアドレスIADDをデータメモリ部33に出力し、リブレース対象のウエイの記憶データDTM[F]を取得し、BCU12に出力するとともに、アドレスADR1に対応するインデックスアドレスIADTをタグメモリ部33に出力する。コントローラ38は更に、リブレース対象のウ

エイのタグアドレスTAM[]を取得し、アドレス信号BADをBCU12に出力し、データメモリ部33の内容を主記憶装置14にコピーバックする。その後、タグメモリ部31内の対応するダーティビットを“0”に書き換える。

【0140】ステップS34では、コントローラ38が、ステップS24で読出したフリーズ対象記憶データBDTをリプレース対象のウエイFのデータメモリ部33に書き込む。

【0141】ステップS35では、CPU10がフリーズ設定命令を実行することにより、コントローラ38が、LRUメモリ部35とフリーズメモリ部36とにアドレスADR1に対応するインデックスアドレスIADLを出力し、LRUメモリ部35に新規アクセスウエイ情報NAW[F]=1を出力する。コントローラ38は更に、フリーズメモリ部36にFRZライトデータFW[F]=1、FRZライトイネーブル信号FE[F]=1を出力することで、フリーズメモリ20のアドレスIADLにFRZ[F]=1を書き込み、ウエイFの該当アドレスをフリーズ状態に設定する。

【0142】フリーズ情報をフリーズメモリ20に書き込む手順としては、図9に示す手順以外にプリセット方式とオンデマンド方式などがある。プリセット方式は、高速実行したい特定のプログラムや高速参照した特定の数値データを、プログラムを起動する段階や特定のプログラムを実行する直前に予めキャッシュメモリ装置11に取り込み、フリーズしておく方式である。具体的には、CPUが主記憶領域の所定のアドレスから記憶データを読み出し、I/O命令で、キャッシュメモリ装置11のデータメモリと、タグメモリと、フリーズメモリとに夫々直接書き込むことにより行われる。

【0143】オンデマンド方式は、高速実行したい特定のプログラムや高速参照した特定の数値データをキャッシュメモリ装置11に取り込み、プログラム実行中にこれをフリーズし、或いは、フリーズ解除する方式である。具体的には、CPUが通常のロード命令により所定のアドレスをアクセスすることで、主記憶領域の所定のアドレスから記憶データが読み出され、キャッシュメモリ装置11にキャッシュされると共に、CPUに渡される。CPUは、上記記憶データがキャッシュされたウエイ情報をキャッシュメモリ装置11から読み出し、このキャッシュされたデータのエントリアドレスと、ウエイに対応するフリーズメモリのビットとに、フリーズ状態又はフリーズ解除状態のデータを書き込む。

【0144】図12は、図1に示したリプレース判定部37の詳細なブロック図である。リプレース判定部37は、フリーズ制御回路21と、LRUデコード回路22、

とで構成される。リプレース判定部37のフリーズ制御回路21及びLRUデコード回路22は夫々論理回路で構成される。なお、フリーズ制御回路21及びLRUデコード回路22を夫々、ソフトウェアのプログラムで実施できるように構成しても良い。この場合、オリジナルのアクセス履歴は、例えばLRUメモリ部35に格納され、中間情報の生成のために用いられる。

【0145】フリーズ制御回路21は、LRUメモリ部35から6ビットの履歴情報W[0,1]~W[2,3]と、3ビットのフリーズ情報FRZ[0]~FRZ[2]とが入力され、6ビットの中間情報M[0,1]~M[2,3]を出力する。

【0146】LRUデコード回路22は、6ビットの中間情報M[0,1]~M[2,3]に基づいてリプレース信号OUT[0]~OUT[3]を出力する。

【0147】ここで、リプレース判定部37の動作の概要を説明する。まず、フリーズ情報FRZ[0]~FRZ[2]が全て“0”で、フリーズ処理を行わない場合のリプレース判定部37の処理について説明する。このとき、M[A,B]=W[A,B]であるので、以下の説明ではW[A,B]を使って説明する。

【0148】LRUデコード回路22は、LRUメモリ部35(図7)から読み出された、履歴情報W[0,1]~W[2,3]の全6ビットに基づいて、最も過去に使われたウエイを検出し、ウエイ0~3に夫々対応するリプレース信号OUT[0]~OUT[3]を出力する。リプレース信号OUT[0]~OUT[3]の値のうちいずれか1つのみが“1”となり、他の値は“0”となる。ビット値が“1”のウエイがリプレース対象となる。

【0149】図4に示したアクセス履歴表から判るように、ウエイ0が最も過去にアクセスされていた場合(最上段のグループを参照)には、履歴情報W[0,1]、W[0,2]、W[0,3]が共に“0”である。ウエイ1が最も過去にアクセスされていた場合(2段目のグループを参照)には、履歴情報W[0,1]が“1”で、W[1,2]及びW[1,3]が“0”である。また、ウエイ2が最も過去にアクセスされていた場合(3段目のグループを参照)には、履歴情報W[0,2]及びW[1,2]が“1”で、W[2,3]が“0”である。ウエイ3が最も過去にアクセスされていた場合(最下段のグループを参照)には、履歴情報W[0,3]及びW[1,3]、W[2,3]が共に“1”である。

【0150】上記関係を式で表現すると次のようになる。即ち、ウエイDに対応する出力値OUT[D]( $0 \leq D \leq 3$ を満たす整数)は、次の各論理式(1)~(4)によって定まる。

$$\text{OUT}[0] = (!W[0,1]) \times (!W[0,2]) \times (!W[0,3]) \quad \cdots(1)$$

$$\text{OUT}[1] = (W[0,1]) \times (!W[1,2]) \times (!W[1,3]) \quad \cdots(2)$$

$$\text{OUT}[2] = (W[0,2]) \times (W[1,2]) \times (!W[2,3]) \quad \cdots(3)$$

$$\text{OUT}[3] = (W[0,3]) \times (W[1,3]) \times (W[2,3]) \quad \dots\dots(4)$$

【0151】本明細書において、例えば「!W」は「W」の反転値(NOT論理)を示す。つまり、先頭部分に「!」を付した数又は符号の値は「!」を付さない対応する数又は符号の反転値を示す。

【0152】例えば、図3に示すように、アクセス履歴が「0←1←2←3」であるとき(同図「ウエイ0ヒット」内の最上段)、ウエイ2がヒットすると、アクセス履歴は「2←0←1←3」になり(同図「ウエイ2ヒット」内の最上段)、履歴情報(W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3])は、(1, 0, 1, 0, 1, 1)となる。この値を上記(1)~(4)に代入すると、リプレース信号OUT[0]~OUT[3]は、

$$\begin{aligned} \text{OUT}[0] &= !1 \times !0 \times !1 = 0, \\ \text{OUT}[1] &= 1 \times !0 \times !1 = 0, \\ \text{OUT}[2] &= 0 \times 0 \times !1 = 0, \\ \text{OUT}[3] &= 1 \times 1 \times 1 = 1 \end{aligned}$$

となり、最も過去にアクセスされたウエイ3がリプレースされることになる。

【0153】次に、図12を参照して、中間情報について詳細に説明する。例えば、N=4の場合に、フリーズ制御前のウエイA及びウエイB間における参照の前後関係を示す“0”又は“1”の中間情報をM[A, B]と表記し、ウエイC(0≤C≤N-2の整数)のフリーズ情報(“0”又は“1”)をFRZ[C]と表記し、ウエイD(0≤D≤Nの整数)に対する出力値であるリプレース信号をOUT[D]と表記する。

【0154】ここでは、フリーズメモリ20(図8)は、N-1ウエイ、つまり3ウエイ分のフリーズ情報FRZ[0]、FRZ[1]及びFRZ[2]のみを有する。これにより、例えば4ウエイの場合に、全てのウエイ

$$M[A, B] = W[A, B] + \text{FRZ}[A] \quad (\text{但し、} 0 \leq A < B = (N-1)) \dots(5)$$

$$M[A, B] = (W[A, B] + \text{FRZ}[A]) \times !\text{FRZ}[B]$$

$$(\text{但し、} 0 \leq A < B < (N-1)) \dots(6)$$

となり、上記(5)、(6)式によって一意にフリーズ論理を決定することができる。これにより、ウエイ数が4以上

$$M[0,1] = (W[0,1] + \text{FRZ}[0]) \times (!\text{FRZ}[1]) \quad (B \neq 3)$$

$$M[0,2] = (W[0,2] + \text{FRZ}[0]) \times (!\text{FRZ}[2]) \quad (B \neq 3)$$

$$M[0,3] = (W[0,3] + \text{FRZ}[0]) \quad (B=3)$$

$$M[1,2] = (W[1,2] + \text{FRZ}[1]) \times (!\text{FRZ}[2]) \quad (B \neq 3)$$

$$M[1,3] = (W[1,3] + \text{FRZ}[1]) \quad (B=3)$$

$$M[2,3] = (W[2,3] + \text{FRZ}[2]) \quad (B=3)$$

となる。

【0159】LRUデコード回路22(図12)は、フリーズ制御回路21からの各中間情報に基づいて、OUT[0]~OUT[3]に夫々対応する“0”、“1”をリプレース信号として出力する。リプレース信号として“0”が出力された場合には「該当するウエイをリプレース対象とせず、フリーズする」ことを意味し、

ウエイをフリーズするとキャッシュメモリ装置が機能できなくなるような不都合を防止している。即ち、全ウエイをフリーズすると、以降のリプレース処理が不可能になるので、少なくとも1つのウエイはリプレース可能にしておく必要がある。従って、後述の場合分けが必要になる。

【0155】LRUメモリ15(図5)に格納された各履歴情報W[A, B]は、任意に指定された2ウエイA、Bのどちらが最近アクセスされたかという内容を保持している。履歴情報W[A, B]が“0”の場合、ウエイAがウエイBより前にアクセスされたことを表し、“1”の場合、ウエイAがウエイBより後にアクセスされたことを表す。

【0156】フリーズメモリ20に格納された各FRZ[C]のビット値であるフリーズ情報は、“1”のとき「該当するウエイCをフリーズする」ことを示し、“0”のときには「該当するウエイCをフリーズしない」ことを示す。

【0157】各履歴情報W[A, B]に対応するビット値“0”、“1”に対し、フリーズメモリ20(図8)からのフリーズ情報FRZ[C]に従ってフリーズ制御回路21(図12)がフリーズ制御した結果である中間情報をM[A, B]と表記する。フリーズ制御回路21は、記憶された履歴情報W[A, B]を修正することなく、該履歴情報W[A, B]とフリーズ情報FRZ[C]とに基づいて演算を行い、フリーズ指定されたウエイCがリプレース対象にならないように、履歴情報W[A, B]を中間情報M[A, B]に変換する。

【0158】ここで、中間情報M[A, B]を正論理で表すと、

に増加した場合でも対処することができる。上記(5)、(6)式にN=4の場合の具体値を夫々代入すると、

“1”が出力された場合には「該当するウエイをリプレース対象とし、フリーズしない」ことを意味する。

【0160】各ウエイDにおけるOUT[D]に夫々対応して出力されるリプレース信号OUT[0]~OUT[3]は、次の各論理式(7)~(10)によって決定される。

$$\text{OUT}[0] = !M[0,1] \times !M[0,2] \times !M[0,3] \quad \dots\dots(7)$$

$$\text{OUT}[1] = M[0,1] \times !M[1,2] \times !M[1,3] \quad \dots\dots(8)$$

$$\text{OUT}[2] = M[0,2] \times M[1,2] \times !M[2,3] \quad \dots\dots(9)$$

$$\text{OUT}[3] = M[0,3] \times M[1,3] \times M[2,3] \quad \dots\dots(10)$$

【0161】ここで、所望のウエイをフリーズする際の実例を説明する。例えば、アクセス履歴が「3←2←0」であるとき(図3の「ウエイ3ヒット」の上から4段目)、ミスヒットが生じてキャッシュメモリが新たな記憶データで書き換えられる場合を考える。フリーズ情報がないと、前述のように、最も過去にアクセスされたウエイ0が新しい記憶データで書き換えられる。

【0162】いま、ウエイ0がフリーズ設定されているとする。このとき、フリーズメモリ20は、フリーズ制御回路21にフリーズ情報FRZ[0]=1、FRZ[1]=0、FRZ[2]=0を送信する。また、履歴情報(W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3])は、図3に示すように、(0, 0, 0, 1, 0, 0)となっている。これら具体値を夫々、ウエイBによって、つまりBが(N-1)であるか否かによって上記(5)、(6)式の対応する方に代入すると、  
 $M[0,1] = (W[0,1] + 1) \times !0 = 1 \times 1 = 1$   
 $M[0,2] = (W[0,2] + 1) \times !0 = 1 \times 1 = 1$   
 $M[0,3] = (W[0,3] + 1) = 1$   
 $M[1,2] = (W[1,2] + 0) \times !0 = 1 \times 1 = 1$   
 $M[1,3] = (W[1,3] + 0) = 0$   
 $M[2,3] = (W[2,3] + 0) = 0$

となる。これは、図3に示すように、アクセス履歴が「0←3←1←2」(ウエイ0ヒット)の上から5段目)に相当する。つまり、LRUメモリ15にはウエイ0が最も過去にアクセスされたことになっているが、中間情報M[A, B]では、ウエイ0が最も最近アクセスされたかのように変換される。

【0163】LRUデコード回路22は、上記結果に基づき、上記式(7)~(10)を用いてリプレース信号OUT[0]~OUT[3]を算出し、以下のように出力する。  
 $\text{OUT}[0] = !M[0,1] \times !M[0,2] \times !M[0,3] = 0$   
 $\text{OUT}[1] = M[0,1] \times !M[1,2] \times !M[1,3] = 0$   
 $\text{OUT}[2] = M[0,2] \times M[1,2] \times !M[2,3] = 1$   
 $\text{OUT}[3] = M[0,3] \times M[1,3] \times M[2,3] = 0$   
 これらのリプレース信号の出力により、ウエイ0の次に古くアクセスされたウエイ2がリプレースされるので、フリーズされているウエイ0が書き換えられることはない。

【0164】次いで、所望のウエイをフリーズする際の別の具体例を説明する。例えば、アクセス履歴が「3←1←2←0」であるとき(図3の「ウエイ3ヒット」の上から4段目)、ミスヒットが生じてキャッシュメモリが新たな記憶データで書き換えられる場合を考える。フリーズ情報がないと、前述したように、最も過去にア

セスされたウエイ0が新しい記憶データで書き換えられることになる。

【0165】いま、ウエイ0及びウエイ1がフリーズ設定されているとする。このとき、フリーズメモリ20は、フリーズ制御回路21にフリーズ情報FRZ[0]=1、FRZ[1]=1、FRZ[2]=0を送信する。また、履歴情報(W[0,1]、W[0,2]、W[0,3]、W[1,2]、W[1,3]、W[2,3])は、図3に示すように、(0, 0, 0, 1, 0, 0)である。これら具体値を、上記(5)、(6)式の対応する方に代入すると、  
 $M[0,1] = (W[0,1] + 1) \times !1 = 1 \times 0 = 0$   
 $M[0,2] = (W[0,2] + 1) \times !0 = 1 \times 1 = 1$   
 $M[0,3] = (W[0,3] + 1) = 1$   
 $M[1,2] = (W[1,2] + 1) \times !0 = 1 \times 1 = 1$   
 $M[1,3] = (W[1,3] + 1) = 1$   
 $M[2,3] = (W[2,3] + 0) = 0$   
 となる。

【0166】上記履歴情報(0, 1, 1, 1, 1, 0)は、図3に示すように、アクセス履歴が「1←0←3←2」に相当する。つまり、LRUメモリ15には「3←1←2←0」の順にアクセスされたことになっているが、中間情報M[A, B]では、ウエイ1とウエイ0が最近アクセスされたかのように変換されることになる。

【0167】LRUデコード回路22は、上記結果に基づき、上記式(7)~(10)を用いてリプレース信号OUT[0]~OUT[3]を算出し、以下のように出力する。  
 $\text{OUT}[0] = !M[0,1] \times !M[0,2] \times !M[0,3] = 0$   
 $\text{OUT}[1] = M[0,1] \times !M[1,2] \times !M[1,3] = 0$   
 $\text{OUT}[2] = M[0,2] \times M[1,2] \times !M[2,3] = 1$   
 $\text{OUT}[3] = M[0,3] \times M[1,3] \times M[2,3] = 0$   
 これらにより、実際にはウエイ0の次にアクセス履歴が古いウエイ2がリプレースされることになるので、フリーズされているウエイ0とウエイ1が書き換えられることはない。

【0168】以上、本実施形態例では、4ウエイの場合を中心に説明したが、Nが2以上の整数であるNウエイ・セット・アソシアティブ・キャッシュの場合には次のようになる。つまり、OUT[X]の一般式は、X=0の場合、

【数4】

$$\text{OUT}[0] = \prod_{j=1}^{N-1} !M[0,j] \quad \dots\dots(11)$$

0<X<N-1の場合、

【数5】



$$\text{OUT}[X] = \prod_{j=0}^{X-1} M[j, X] \times \prod_{j=X+1}^{N-1} M[X, j] \quad \text{.....(12)}$$

X=N-1の場合、

【数6】

$$\text{OUT}[N-1] = \prod_{j=0}^{N-1} M[j, N-1] \quad \text{.....(13)}$$

となる。ここで、記号 $\Pi$ は、関数 $F(X)$ に対して以下のように定義される。

【数7】

$$\prod_{x=0}^N F(X) = F(0) \times F(1) \times \dots \times F(N-1) \times F(N)$$

【0169】式(11)～(13)を具体的に記述すると次のようになる。

$$\text{OUT}[0] = M[0,1] \times M[0,2] \times \dots \times M[0,N-1]$$

$$\text{OUT}[1] = M[0,1] \times M[1,2] \times \dots \times M[1,N-1]$$

$$\text{OUT}[2] = M[0,2] \times M[1,2] \times \dots \times M[0,N-1]$$

$$\text{OUT}[N-1] = M[0,N-1] \times M[1,N-1] \times \dots \times M[N-2,N-1]$$

【0170】以上の演算を行うことで、いずれか1つのリプレース信号 $\text{OUT}[R]$ が“1”となり、そのウエイRがリプレース対象のウエイと判定されることになる。

【0171】なお、本実施形態例では、正論理を用いて説明したが、これに限らず負論理を用いることもできる。また、記憶された情報に基づいてLRU自身がLRU情報を得られれば良いので、アクセスした情報を必ずしもLRU方式で記憶しなくても良い。

【0172】以上のように、本実施形態例では、LRUメモリ15における格納状態を表す方法として、任意に指定した2つのウエイ間の情報のみを保持する方式を利用し、この方式にフリーズ論理を組み合わせた。従って、極めて小規模な回路構成の追加でフリーズ論理を実現することができる。

【0173】本実施形態例では、ユーザがフリーズアドレス出力回路19に対して設定を行うことにより所望のウエイをフリーズできるので、例えば、利用頻度は少ないが、高速に処理しなければならないプログラムやデータをフリーズした状態で実行又は演算を行うことができる。このため、重要な処理プログラムの実行速度が低下することがない。

【0174】また、従来は1ウエイのみをフリーズ対象としていたが、本実施形態例では、Nウエイのうちの1ウエイのみをリプレース対象として残しておくことにより、他のウエイを全てフリーズ対象とすることもできる。これにより、比較的サイズの大きいプログラムをフリーズすることが可能になり、従来に比してミスヒット

の確率が低減する。その場合、キャッシュの書き換え回数が大幅に減少する。

【0175】本実施形態例では、LRUメモリ33a～33dには常に最新のアクセス履歴情報が保持されており、データメモリ部33のリプレースが必要になったときには、フリーズ情報とアクセス履歴情報とに基づいてリプレース信号を演算によって求め、フリーズ対象を除いたウエイで、一番過去にアクセスがあったウエイをリプレース対象のウエイとして出力する構成とした。このため、本願方式によれば、フリーズ機能を有しながらも、LRU方式の有効性を損なうことなくキャッシュメモリ装置を実現することができる。

【0176】また、フリーズメモリ20は、フリーズ情報を1エントリに(N-1)ビットで記憶しており、必要なメモリ量が少ないので、フリーズメモリ20をRAMではなくレジスタから構成することも可能になる。フリーズメモリ20のメモリ容量は、4ウエイの場合に3ビット×エントリ数であるので、例えば、1ウエイに64個のエントリが存在する際には、192ビットのメモリ量があれば足りる。このため、フリーズを実行するためのロジックが極めて小規模で足りるので、必要な回路構成を簡素化することができる。また、従来例のように、フリーズ情報に基づいてLRUメモリを書き換えることがないので、従来の書換えに要していた時間がなくなり、キャッシュメモリ装置の負荷が軽減されるとともに、キャッシュメモリ装置の応答時間が短縮できる。

【0177】本実施形態例では、フリーズ機能を使わない場合には、フリーズメモリ20の記憶内容を全て“0”にするだけで、従来と同じキャッシュメモリ装置11として機能させることができる。また、フリーズ動作を終了したときには、それまでのアクセス履歴がフリーズ情報で書き換えられることなくLRUメモリ部35に記憶されているので、LRU方式の機能を損なうことがない。このため、ミスヒットが生じた場合、最も過去にアクセスされたウエイがリプレース対象になり、キャッシュメモリのヒット確率が低減することがない。

【0178】本実施形態例におけるキャッシュメモリ装置11では、上記効果に加えて次のような効果をも奏する。例えば、1つのサブルーチンを1ウエイ相当のメモリに格納しようとしても、データ量が多くて格納できない場合がある。この場合、従来例では、1ウエイのみを活用する構成であるため、ウエイを2つ跨ってデータ格納することはできなかった。しかし、本実施形態例では、N-1の複数ウエイを活用することができるので、比較的規模の大きいデータでも容易に格納することができる。

【0179】以上、本発明をその好適な実施形態例に基

づいて説明したが、本発明のキャッシュメモリ装置及びその制御方法は、上記実施形態例の構成にのみ限定されるものではなく、上記実施形態例の構成から種々の修正及び変更を施したキャッシュメモリ装置及びその制御方法も、本発明の範囲に含まれる。

#### 【0180】

【発明の効果】以上説明したように、本発明のキャッシュメモリ装置及びその制御方法によると、簡素な回路構成を備えながらも、キャッシュメモリのスループットを低下させることなく2以上のウエイを対象に円滑なフリーズ処理が実行でき、ニーズに対応したキャッシュメモリ機能を実現することができる。マイクロコンピュータの用途は拡大しており、低価格化が要求されている。しかし、用途毎にマイクロコンピュータを開発していたのでは、低価格化は実現できない。本願発明のように、フリーズ機能付きのキャッシュメモリ装置を採用することで、同一構成のマイクロコンピュータで色々な用途に対応することが可能になる。例えば、フリーズ機能なしのキャッシュメモリ装置、フリーズ機能付きのキャッシュメモリ装置、或いは、キャッシュメモリ装置の一部を高速読出し専用メモリ（ROM）または高速RAMとして利用したマイクロコンピュータ・システムなどに対応することができる。従って、顧客の多様な要求に対して、ハードウェアを設計変更することなく、ソフトの変更だけで応じることが可能になり、短期間で低価格のマイクロコンピュータを顧客に提供することが可能になる。

【0181】また、フリーズ情報に対応してアクセス履

$$OUT[X] = \prod_{j=0}^{X-1} M[j, X] \times \prod_{j=X+1}^{N-1} M[X, j]$$

(但し、 $0 \leq X, j \leq (N-1)$ 、 $[,]$  内の数字で左側の数<右側の数)

に基づいて行うことにより、中間情報を生成することによって、従来のリプレース対象ウエイの算出回路をそのまま利用することが可能になる。

#### 【図面の簡単な説明】

【図1】本発明に係るキャッシュメモリ装置を備えたコンピュータシステムの一実施形態例を示すブロック図である。

【図2】Nウエイ分のアクセス履歴を示す模式図であり、(a)はN=2の場合、(b)はN=3の場合、(c)はN=4の場合、(d)はN=5の場合を夫々示す。

【図3】4ウエイの全てのアクセス順序と履歴情報との関係を、ヒットしたウエイを基準に示した履歴情報図である。

【図4】4ウエイの全てのアクセス順序と履歴情報との関係を、最も古くにアクセスされたウエイを基準に示した履歴情報図である。

【図5】本実施形態例におけるタグメモリ部及びその周

辺情報を中間情報に変換した後にリプレース対象のウエイを算出することにより、既に設計済みのキャッシュメモリ装置の資産を継承しつつ、簡単な論理回路を付加するだけで、フリーズ機能付きのキャッシュメモリ装置が実現できる。この場合、LRUの元データを破棄することがないので、フリーズ状態を解除したとき、それまでのアクセス履歴に基づいた通常のLRU方式に即座に復帰できる。

【0182】更に、Nウエイのアクセス順序を最近アクセスされたウエイから順に並べたとき、フリーズ対象のウエイが1乃至N-1のいずれかのアクセス順序になるような中間情報を生成してからリプレース対象のウエイを算出すると、LRUの機能を損なうことなくリプレース対象のウエイ算出することができる。

【0183】また、中間情報としてのM[A, B]への変換を、

$$M[A, B] = W[A, B] + FRZ[A]$$

$$M[A, B] = (W[A, B] + FRZ[A]) \times !FRZ[B]$$

に基づいて行うことにより、簡単な論理で、フリーズ対象のウエイがリプレース対象のウエイにならないように変換処理することができる。これにより、小規模な論理回路を追加するだけでフリーズ機能付きのキャッシュメモリ装置を実現することができる。

【0184】更に、リプレース対象のウエイXの算出を、次式

【数8】

辺の構成を示すブロック図である。

【図6】本実施形態例におけるデータメモリ部及びその周辺の構成を示すブロック図である。

【図7】本実施形態例におけるLRUメモリ部及びその周辺の構成を示すブロック図である。

【図8】本実施形態例におけるフリーズメモリ部及びその周辺の構成を示すブロック図である。

【図9】本実施形態例におけるフリーズI/O命令によるフリーズ設定手順を示す流れ図である。

【図10】本実施形態例におけるフリーズ設定手順を説明するための主記憶装置のアドレス配置例を示す図である。

【図11】本実施形態例におけるフリーズ設定命令によるフリーズ設定手順を示す流れ図である。

【図12】本実施形態例におけるリプレース判定部及びその周辺の構成を示すブロック図である。

【図13】従来の半導体記憶装置に対応するキャッシュメモリ装置の構成を示すブロック図である。

【図14】従来のLRUパージ/フリーズ制御部のフリーズ処理を示す流れ図である。

【図15】従来の各記憶データの優先順位を考慮したパージ/フリーズ制御の概要を示す図である。

【符号の説明】

10: CPU

11: キャッシュメモリ装置

12: バスコントロールユニット(BCU)

13: システムバス

14: 主記憶装置

15: LRUメモリ

16: LRU書込みデータ生成回路

19: フリーズアドレス出力回路

20: フリーズメモリ

21: フリーズ制御回路

22: LRUデコード回路

31: タグメモリ部

31a~31d: タグメモリ

32: 比較部

32a~32d: 比較回路

33: データメモリ部

33a~33d: データメモリ

34: セレクタ

35: LRUメモリ部

36: フリーズメモリ部

37: リプレイス判定部

61: TAGアドレスデコーダ

62: TAG書込み制御回路

64: DMアドレスデコーダ

65、68: セレクタ

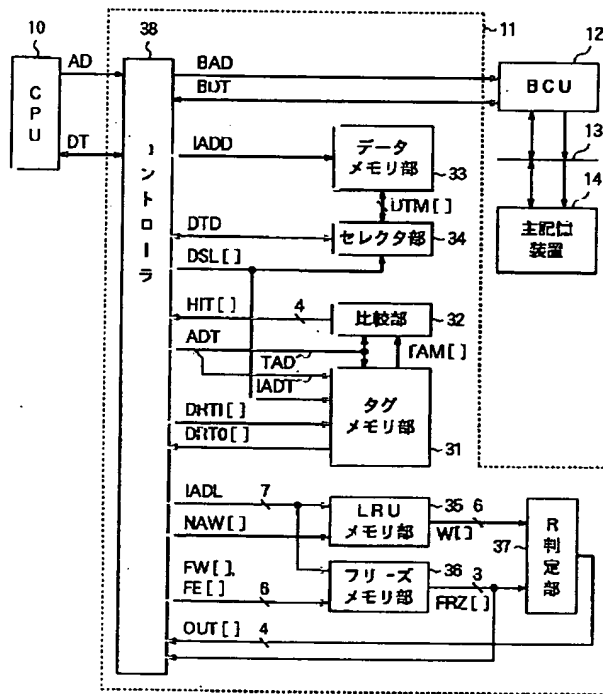
66: LRUアドレスデコーダ

67: LRU書込み制御回路

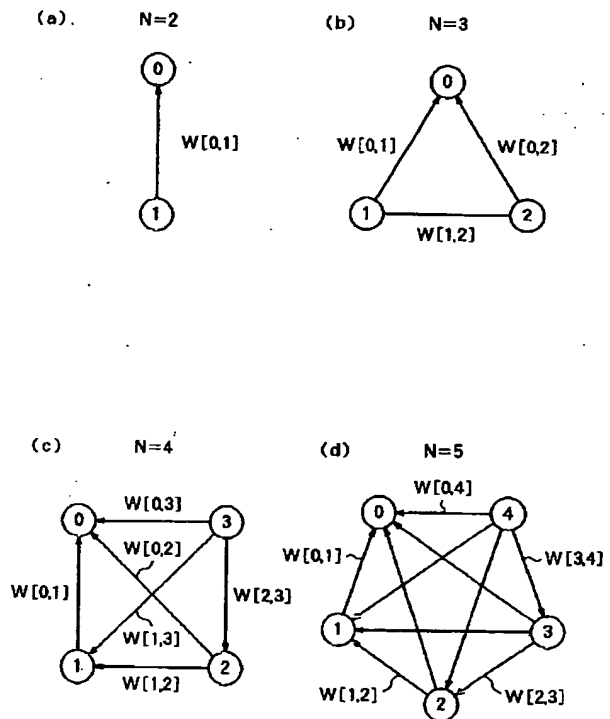
69: FRZアドレスデコーダ

70: FRZ書込み制御回路

【図1】



【図2】



【図3】

履歴情報

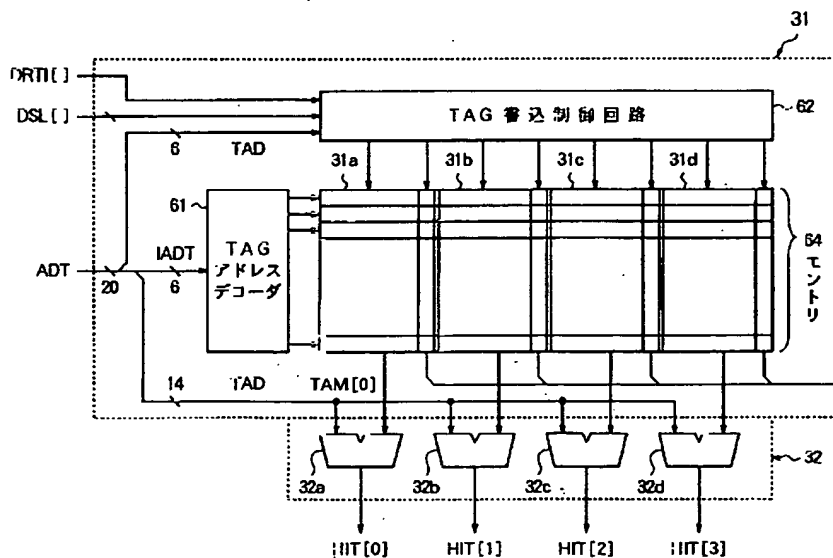
	t <sub>4</sub> -t <sub>3</sub> -t <sub>2</sub> -t <sub>1</sub>	W[0,1]	W[0,2]	W[0,3]	W[1,2]	W[1,3]	W[2,3]
ウェイ0ヒット	0 1 2 3	1	1	1	1	1	1
	0 1 3 2	1	1	1	1	1	0
	0 2 1 3	1	1	1	0	1	1
	0 2 3 1	1	1	1	0	0	1
	0 3 1 2	1	1	1	1	0	0
	0 3 2 1	1	1	1	0	0	0
ウェイ1ヒット	1 0 2 3	0	1	1	1	1	1
	1 0 3 2	0	1	1	1	1	0
	1 2 0 3	0	0	1	1	1	1
	1 2 3 0	0	0	0	1	1	1
	1 3 2 0	0	1	0	1	1	0
	1 3 2 0	0	0	0	1	1	0
ウェイ2ヒット	2 0 1 3	1	0	1	0	1	1
	2 0 3 1	1	0	1	0	0	1
	2 1 0 3	0	0	1	0	1	1
	2 1 3 0	0	0	0	0	1	1
	2 3 0 1	1	0	0	0	0	1
	2 3 1 0	0	0	0	0	0	1
ウェイ3ヒット	3 0 1 2	1	1	0	1	0	0
	3 0 2 1	1	1	0	0	0	0
	3 1 0 2	0	1	0	1	0	0
	3 1 2 0	0	0	0	1	0	0
	3 2 0 1	1	0	0	0	0	0
	3 2 1 0	0	0	0	0	0	0

【図4】

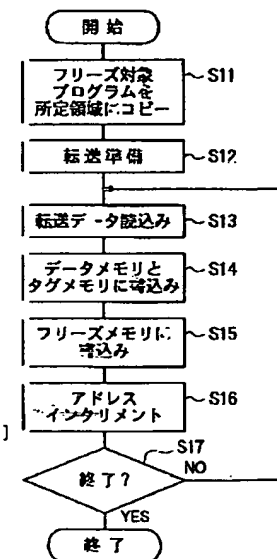
履歴情報

	t <sub>4</sub> -t <sub>3</sub> -t <sub>2</sub> -t <sub>1</sub>	W[0,1]	W[0,2]	W[0,3]	W[1,2]	W[1,3]	W[2,3]
OUT[0]	3 2 1 0	0	0	0	0	0	0
	2 3 1 0	0	0	0	0	0	1
	3 1 2 0	0	0	0	0	1	0
	1 3 2 0	0	0	0	1	1	0
	2 1 3 0	0	0	0	0	1	1
	1 2 3 0	0	0	0	1	1	1
OUT[1]	3 2 0 1	1	0	0	0	0	0
	2 3 0 1	1	0	0	0	0	0
	3 0 2 1	1	1	0	0	0	1
	0 3 2 1	1	1	1	0	0	0
	2 0 3 1	1	0	1	0	0	1
	0 2 3 1	1	1	1	0	0	1
OUT[2]	3 1 0 2	0	1	0	1	0	0
	1 3 0 2	0	1	0	0	1	0
	3 0 1 2	1	1	0	1	0	0
	0 3 1 2	1	1	1	1	0	0
	1 0 3 2	0	1	1	1	1	0
	0 1 3 2	1	1	1	1	1	0
OUT[3]	2 1 0 3	0	0	1	0	1	1
	1 2 0 3	0	0	1	1	1	1
	2 0 1 3	1	0	1	0	1	1
	0 2 1 3	1	1	1	0	1	1
	1 0 2 3	0	1	1	1	1	1
	0 1 2 3	1	1	1	1	1	1

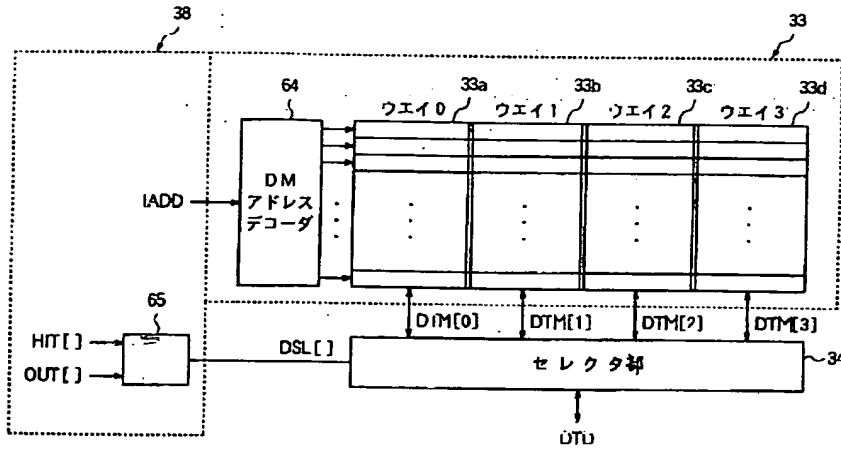
【図5】



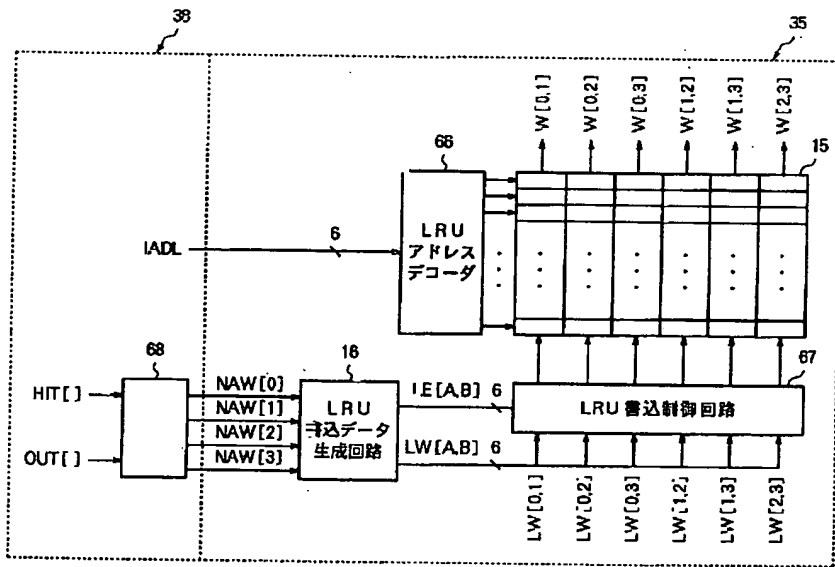
【図9】



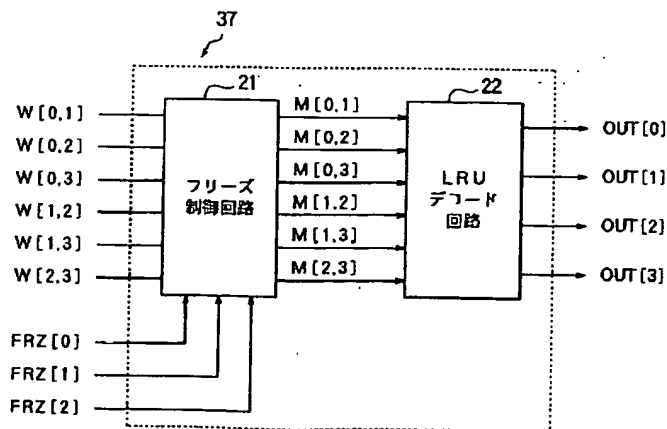
【図6】



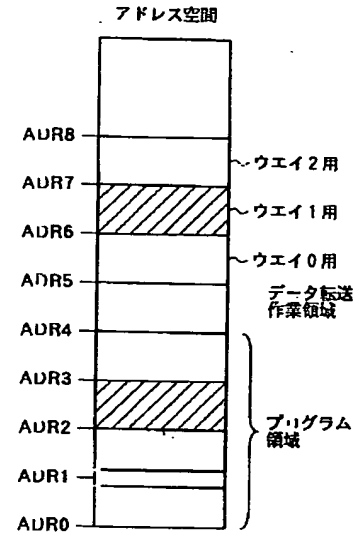
【図7】



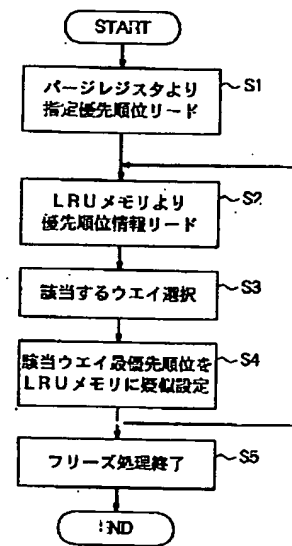
【図12】



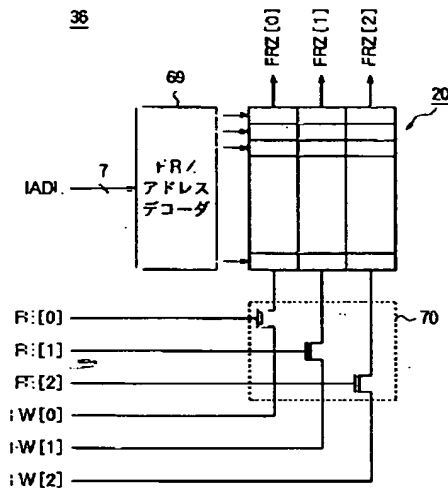
【図10】



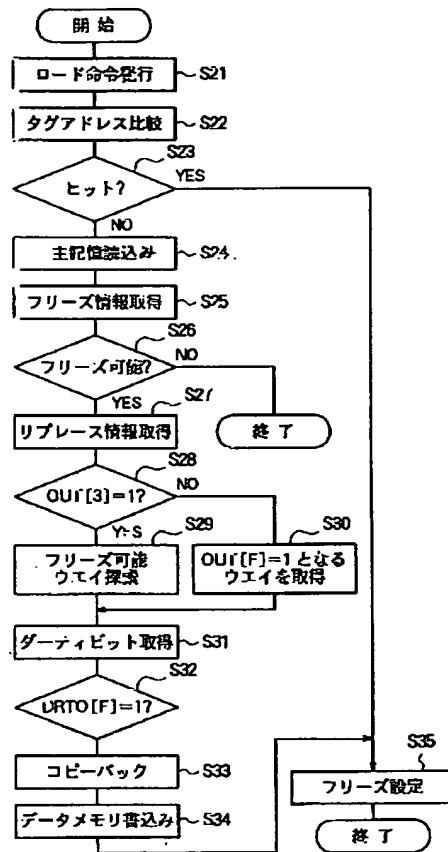
【図14】



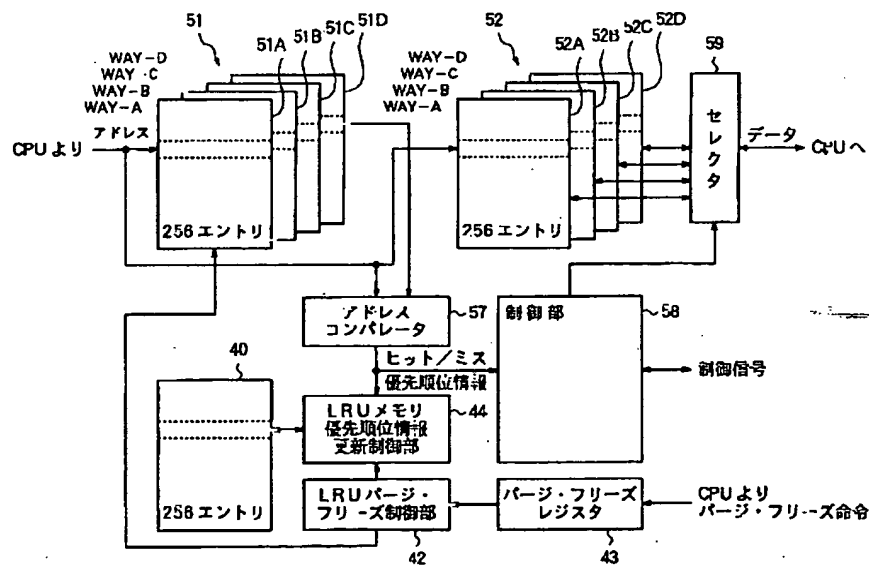
【図8】



【☒ 1 1】



【图13】



【図15】

